

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN, EVALUACIÓN Y COMPARACIÓN DE ALGORITMOS DE CLASIFICACIÓN SUPERVISADOS PARA IDENTIFICAR A PROPIETARIOS DE FIRMAS MANUSCRITAS

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER
EN CIENCIAS DE LA COMPUTACIÓN MENCIÓN EN INTELIGENCIA ARTIFICIAL**

ING. GERMAN FRANCISCO CULQUI CULQUI

german.culqui@epn.edu.ec

Directora: PhD. Sandra Patricia Sánchez Gordón

sandra.sanchez@epn.edu.ec

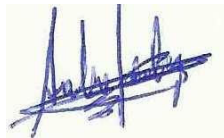
Codirectora: PhD. Myriam Hernández

myriam.hernandez@epn.edu.ec

Quito, Noviembre del 2020

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación IMPLEMENTACIÓN Y ANÁLISIS COMPARATIVO DE ALGORITMOS DE CLASIFICACIÓN SUPERVISADA PARA VERIFICACIÓN DE FIRMAS MANUSCRITAS desarrollado por Germán Francisco Culqui Culqui, estudiante de la Maestría en Ciencias de la Computación con Mención en Inteligencia Artificial , habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.



PhD. Sandra Patricia Sánchez

DIRECTOR



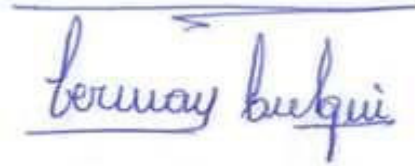
PhD. Myriam Hernández

CODIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Germán Francisco Culqui Culqui, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Ing. Germán Francisco Culqui Culqui

DEDICATORIA

A mi hija Gémary Culqui por ser mi fuente de inspiración y la luz que ilumina el camino a seguir ante cualquier adversidad que se presenta en mi vida.

A mis padres Mercedes y Fernando Culqui, que siempre me apoyaron en todo momento, enseñándome a ser un buen hijo, un mejor ciudadano, buen padre y una buena persona para la sociedad ecuatoriana.

A mis hermanos Borys, Edwin y Fernando con quienes compartí mi niñez, adolescencia y juventud, ellos son mi familia que me ha enseñado que son las personas en las que siempre podré confiar frente a cualquier problema.

AGRADECIMIENTO

Mis más sinceros agradecimientos para la Dra. Sandra Sánchez y Dra. Myriam Hernández, directora y codirectora respectivamente, quienes han sido guía y apoyo en este trabajo de investigación.

A la Escuela Politécnica Nacional por el apoyo incondicional en la provisión de la beca de estudio, que permitió reducir los costos de las matrículas y de esta manera cursar esta interesante maestría.

A los profesores de la carrera que, gracias a su guía, enseñanza y paciencia, enrumbaron mi deseo de investigación.

A todas las personas que me brindaron su confianza colaborando de manera desinteresada al entregar ejemplares de su firma, en ayudarme en la recolección de firmas, para la realización de este proyecto. A todas las personas les reitero mi compromiso de cuidar y mantener el sigilo y la reserva de la firma entregada, asegurando el uso exclusivo en el desarrollo de este proyecto, cuidando que los datos no sean entregados a terceras personas debido a lo delicado de la información.

TABLA DE CONTENIDO

TABLA DE CONTENIDO	vi
LISTA DE FIGURAS	ix
LISTA DE TABLAS	xiii
LISTA DE ECUACIONES	xv
ABSTRACT	xvii
1. INTRODUCCIÓN	18
1.1. ANTECEDENTES.....	18
1.2. PLANTEAMIENTO DEL PROBLEMA	20
1.3. OBJETIVO GENERAL	20
1.4. OBJETIVOS ESPECÍFICOS.....	20
1.5. MARCO TEÓRICO	21
1.5.1. Reconocimiento de la firma manuscrita como sistema biométrico	21
1.5.2. Pre procesamiento de una imagen digital	22
1.5.3. Histograma de una imagen.....	23
1.5.4. Ecuación del histograma de una imagen.....	24
1.5.5. Métodos de eliminación de ruido en imágenes	26
1.5.6. Esqueletización de imágenes en tonos de gris	28
1.5.7. Características de una imagen (descriptores).....	30
1.5.8. Redes neuronales	31
1.5.8.1. Función escalón.....	33
1.5.8.2. Funciones lineal y mixta	34
1.5.8.3. Función tangente hiperbólica	34
1.5.8.4. Función sigmoidea	35
1.5.8.5. Redes neuronales convolucionales	36

1.5.9.	Reconocimiento de patrones y clasificación	40
1.5.10.	Métricas de Evaluación	43
2.	REVISIÓN DE LITERATURA.....	48
2.1.	PREGUNTAS DE INVESTIGACIÓN.....	48
2.2.	ESTRATEGIA DE BÚSQUEDA.	48
2.3.	DEFINICION DE BASES DE DATOS Y MOTORES DE BÚSQUEDA.	49
2.3.1.	Definición de criterios de inclusión y exclusión	49
2.3.2.	Búsqueda y clasificación de estudios relevantes.	50
2.3.3.	Extracción y síntesis de datos.	51
3.	MATERIALES Y MÉTODOS.....	57
3.1.	ADQUISICIÓN DE DATOS.....	58
3.2.	SELECCIÓN, ADAPTACIÓN E IMPLEMENTACIÓN DE ALGORITMOS.	62
3.2.1.	Selección de herramientas de desarrollo.....	62
3.2.2.	Selección de algoritmos de clasificación.....	63
3.2.3.	Pre proceso de la imagen de una firma.	69
3.3.	Entrenamiento y validación de algoritmos.....	72
4.	RESULTADOS Y DISCUSION.	74
4.1.	RESULTADOS DEL ENTRENAMIENTO.....	74
4.1.1.	Resultados del entrenamiento del algoritmo CN-GC, base CEDAR.....	74
4.1.2.	Datos de entrenamiento del algoritmo CN-GC, dataset DB-GC.....	79
4.1.3.	Entrenamiento del algoritmo Dígitos, base CEDAR.	85
4.1.4.	Entrenamiento algoritmo Dígitos, base DB-GC.....	89
4.1.5.	Entrenamiento red convolucional poco profunda, base CEDAR.	95
4.1.6.	Entrenamiento red convolucional poco profunda, base DB-GC.	100
4.2.	RESULTADOS DE LA ETAPA DE PRUEBAS.....	106
4.3.	COMPARATIVO Y DISCUSIÓN DE RESULTADOS.	112

5. CONCLUSIONES Y RECOMENDACIONES.....	121
5.1. CONCLUSIONES.....	121
5.2. RECOMENDACIONES.....	122
REFERENCIAS	124

LISTA DE FIGURAS

Figura 1. Ejemplo de una firma manuscrita.	21
Figura 2. Ejemplo del ruido no visible de la imagen de una firma.	22
Figura 3. Histograma de una imagen en color.	23
Figura 4. Histograma de una imagen en tonos de gris.	23
Figura 5. Imagen gris original para ser procesada usando ecualización.	24
Figura 6. Imagen mejorada usando ecualización.	25
Figura 7. Imagen de un paisaje después del proceso de ecualización.	25
Figura 8. Imagen ecualizada y su histograma	26
Figura 9. Operación de convolución en una imagen gris.	27
Figura 10. Triangulo de Pascal para generación de coeficientes gaussianos.	27
Figura 11. Ejemplo de filtros para usos diversos.	28
Figura 12. Notación usada en el algoritmo de esqueletización.	28
Figura 13. Ejemplo de cambios de 0 a 1 y de 1 a 0.	29
Figura 14. Imagen de una firma original y el esqueleto de la imagen.	30
Figura 15. Diagrama de un nodo o neurona (ecuación 3)	32
Figura 16. Función escalonada [20].	33
Figura 17. Funciones lineales mixtas [20].	34
Figura 18. Función tangente hiperbólica [20].	35
Figura 19. Función sigmoideal [20].	35
Figura 20. Arquitectura de una red neuronal convolucional.	36
Figura 21. Operación de convolución en imágenes de tonos de gris.	37
Figura 22. Imagen resultante sin relleno (padding)	37
Figura 23. Operación de relleno (padding)	38
Figura 24. Operación de pooling en un área de 3x3, valor máximo.	38
Figura 25. Operación de pooling en un área de 2x2, valor máximo.	39

Figura 26. Ejemplo de stacking de una imagen.....	39
Figura 27. Ejemplo de imagen con stride = 1	40
Figura 28. Ejemplo imagen con stride = 2	40
Figura 29. Clasificación en base a las características [15].	41
Figura 30. Clasificador lineal [15]	42
Figura 31. Clasificadores no lineales. [15].....	42
Figura 32. Matriz de confusión [25].	43
Figura 33. Publicaciones por base de datos.....	50
Figura 34. Publicaciones agrupadas por año de publicación.....	51
Figura 35. Extracción de la característica del ángulo [29].	53
Figura 36. Transformada discreta de Radón de dos firmantes.	54
Figura 37. Red generativa adversaria.	55
Figura 38. Red convolucional siamesa [36].....	56
Figura 39. Esquema general de un clasificador de imágenes	58
Figura 40. Formato de recolección de firmas ineficiente.	59
Figura 41. Problema de firma incompleta debido al uso de un formato libre.	60
Figura 42. Formato final de recolección de firmas.....	60
Figura 43. Firmantes por rango de edad de la base de firmas DB-GC.....	61
Figura 44. Estructura de directorios para las imágenes de firmas.	72
Figura 45. Exactitud de entrenamiento y validación, algoritmo CN-GC, CEDAR.....	75
Figura 46. EMC, entrenamiento y validación, algoritmo CN-GC, dataset CEDAR.....	76
Figura 47, LOSS, entrenamiento y validación, algoritmo CN-GC, dataset CEDAR.	76
Figura 48. Matriz de confusión, primer algoritmo, para la base de datos CEDAR.	78
Figura 49. Exactitud de validación y entrenamiento, algoritmo CN-GC, base DB-GC.	80
Figura 50. EMC de validación y entrenamiento, algoritmo CN-GC, base DB-GC.....	80
Figura 51. LOSS de validación y entrenamiento, algoritmo CN-GC, base DB-GC.	81

Figura 52. Matriz de confusión del algoritmo CN-GC, base de datos DB-GC.	84
Figura 53. Exactitud de entrenamiento, CNN reconocimiento de dígitos, base CEDAR. .	86
Figura 54. EMC de entrenamiento, CNN reconocimiento de dígitos, base CEDAR.....	86
Figura 55. LOSS de entrenamiento, CNN reconocimiento de dígitos, base CEDAR.	87
Figura 56. Matriz de confusión, algoritmo reconocimiento de dígitos, base CEDAR.....	89
Figura 57. Exactitud de entrenamiento, segundo algoritmo, base DB-GC.....	91
Figura 58. EMC de entrenamiento, CNN reconocimiento de dígitos, base DB-GC.....	91
Figura 59. LOSS de entrenamiento, CNN reconocimiento de dígitos,, base DB-GC.	92
Figura 60. Matriz de confusión, algoritmo Dígitos, base DB-GC.....	95
Figura 61. Exactitud de entrenamiento, tercer algoritmo, base CEDAR	96
Figura 62. EMC de entrenamiento, tercer algoritmo, base CEDAR	97
Figura 63. Loss de entrenamiento, tercer algoritmo, base CEDAR.	97
Figura 64. Matriz de confusión, tercer algoritmo, base CEDAR.....	99
Figura 65. Exactitud, tercer algoritmo, base DB-GC.....	101
Figura 66. EMC, tercer algoritmo, base DB-GC.	101
Figura 67. LOSS de entrenamiento, tercer algoritmo, base DB-GC.	102
Figura 68. Matriz de confusión, tercer algoritmo, base DB-GC.....	105
Figura 69. Comparativo métrica de exactitud/accuracy.	108
Figura 70. Comparativo de métrica de precisión.	108
Figura 71. Comparativo de tasa de falsos positivos (TFP).	109
Figura 72. Comparativo métrica de sensibilidad.....	110
Figura 73. Comparativo métrica de especificidad.....	110
Figura 74. Comparativo F1-Score.	111
Figura 75. Histograma de frecuencias y distribución acumulada de precisión, CEDAR.	112
Figura 76. Histograma de frecuencias y distribución acumulada de Precisión. DB-GC.	114
Figura 77. F1-Score, base de datos CEDAR.....	115

Figura 78. F1-Score, base de datos DB-GC.....	116
Figura 79. Comparativo tasa de falsos positivos, base CEDAR.	118
Figura 80. Comparativo de tasa de falsos positivos, base DB-GC.	119

LISTA DE TABLAS

Tabla 1. Métricas de evaluación para las redes neuronales [26].....	44
Tabla 2. Matriz de confusión multiclase [26].	45
Tabla 3. Matriz de confusión multiclase, definición de valores VP, VN, FP y FN.	46
Tabla 4. Resultados después de aplicar los criterios de inclusión y exclusión.	50
Tabla 5. Descripción red convolucional CN-GC.....	65
Tabla 6. Estructura de la red convolucional reconocimiento dígitos [39]......	67
Tabla 7. Red neuronal convolucional poco profunda.	69
Tabla 8. Datos de entrenamiento, algoritmo CN-GC, base CEDAR esqueletizada.	70
Tabla 9. Datos de entrenamiento, tercer algoritmo con la base CEDAR esqueletizada. ..	70
Tabla 10. Matriz de mediana modificada, papel de gramaje delgado.	71
Tabla 11. Matriz de mediana normal para eliminación de ruido.	71
Tabla 12. Datos del entrenamiento, algoritmo CN-GC, base de firmas CEDAR.	75
Tabla 13. Resultados fase de prueba, algoritmo CN-GC, base de datos CEDAR.....	78
Tabla 14. Datos de entrenamiento algoritmo CN-GC, base de firmas DBGC.	79
Tabla 15. Datos fase de validación, algoritmo CN-GC, base DB-GC.....	83
Tabla 16. Datos de entrenamiento, CNN reconocimiento dígitos, base CEDAR.....	85
Tabla 17, Resultados fase de validación, segundo algoritmo, base de datos CEDAR.	88
Tabla 18. Datos entrenamiento, algoritmo dos, base DB-GC.	90
Tabla 19. Resultados fase de prueba, algoritmo Dígitos, base de datos DB-GC.	94
Tabla 20. Datos de entrenamiento, algoritmo poco profundo, base CEDAR.....	96
Tabla 21. Resultados fase de prueba, tercer algoritmo, base de datos CEDAR.	99
Tabla 22. Datos de entrenamiento, tercer algoritmo, base DB-GC.	100
Tabla 23. Resultados fase de prueba, tercer algoritmo, base de datos DB-GC.	104
Tabla 24. Comparativo en la fase de entrenamiento.....	106
Tabla 25. Comparativo general de los algoritmos, base de firmas CEDAR.	107

Tabla 26. Comparativo general de los algoritmos, base de firmas DB-GC.	107
Tabla 27. Datos de frecuencia y distribución acumulada de precisión base CEDAR.	112
Tabla 28. Datos de frecuencia y distribución acumulada de precisión base DB-GC.	113
Tabla 29. Clases con F1-score menor a 0.7, algoritmo CN-GC, base DB-GC.	116
Tabla 30. Clases con F1-score menor a 0.7, algoritmo P. Prof, base DB-GC.	117
Tabla 31. Tasa de falsos positivos base CEDAR.....	118
Tabla 32. Tasa de falsos positivos base DB-GC.....	119

LISTA DE ECUACIONES

Ecuación 1. Ecuación de ecualización	24
Ecuación 2. Producto de entradas y pesos más una constante	32
Ecuación 3. Estructura matemática de una neurona.....	32
Ecuación 4. Operación de convolución de matrices.....	37
Ecuación 5. Error medio cuadrático	46

RESUMEN

La firma manuscrita usada para la identificación personal es obligatoria en documentos legales, cheques bancarios, contratos, tarjetas de crédito, entre otros documentos públicos y privados. En especial las instituciones financieras, evitan y controlan los errores que se pueden cometer en la verificación de la firma manuscrita por parte del personal operativo en documentos que involucran transacciones de dinero, tales como papeletas de retiros, cheques, pagarés, etc. Estas instituciones realizan grandes inversiones en sistemas que les permitan identificar la firma de una persona en tiempos muy pequeños, debido a la gran cantidad de documentos a validar. En este trabajo se propone evaluar tres modelos de redes neuronales convolucionales que permitan clasificar e identificar la firma de una persona con una exactitud superior al 88%, de manera rápida y eficaz, uno de los modelos usados fue desarrollado por el autor. Se muestra que al menos dos de los modelos usados superan la meta establecida. Para realizar esta evaluación y comparación se usaron dos bases de datos de firmas, la primera de nombre CEDAR de acceso público y la segunda una base de 121 firmas recolectadas por el autor de este trabajo con firmas locales.

Palabras clave: Clasificación de firmas manuscritas, Algoritmos de clasificación supervisados, Redes Neuronales convolucionales, Verificación off-line de firmas manuscritas.

ABSTRACT

The handwritten signature used for personal identification is mandatory in legal documents, bank checks, credit cards, and contracts, among other public and private documents. For financial institutions it is essential to avoid and control errors that can be made in verifying the handwritten signature by operating personnel on documents that involve money transactions, such as withdrawal slips, checks and promissory notes. These institutions make large investments in systems that identify a person's signature in very short times due to a large number of documents to be validated. In this work, it is proposed to evaluate three models of convolutional neural networks that can classify and identify a person's signature. The author of the present thesis developed one of the models used, and it presents an accuracy greater than 88%. At least two of the used models are shown to exceed the set of goals. Two databases of signatures were used to carry out this evaluation and comparison, the first one called CEDAR publicly available and the second a signature base collected by the author of this work with local handwritten signatures.

Keywords: Classification of handwritten signatures, supervised classification algorithms, Convolutional Neural Networks, Off-line verification of handwritten signatures.

1. INTRODUCCIÓN

1.1. ANTECEDENTES

La firma manuscrita es la más aceptada para la identificación personal, su inclusión es obligatoria en documentos legales, cheques bancarios, contratos, tarjetas de crédito, entre otros documentos públicos y privados. La falsificación de una firma es un objetivo atractivo para el fraude y constituye uno de los métodos más comunes para suplantar la identidad de una persona.

Para salvaguardar los intereses de una organización es necesario evitar estos eventos de falsificación malintencionados; por ejemplo, en una entidad estatal o en instituciones financieras. En las instituciones financieras, es muy importante evitar y controlar la falsificación en la validación de las transacciones realizadas mediante soportes físicos, haciendo comprobación de la firma en documentos, tales como papeletas de retiros, cheques, notas de débito, entre otros [1, 2], en las cuales, la firma es susceptible a ser falsificada. Por ello, es necesario el uso de sistemas de verificación que permitan la validación de la identidad de un cliente comparando su firma con la existente en una base de datos de firmas registradas.

Uno de los principales procedimientos que existe en Ecuador es el pago de los cheques depositados en las entidades financieras a través de la Cámara de Compensación de Cheques (CCC), coordinada por el Banco Central. En el proceso, el gran número de cheques recibidos cotidianamente, son sometidos a una revisión visual de la firma manuscrita en el documento con la registrada en los archivos de cada institución [3, 2, 1]. Este proceso de verificación de firmas sobre una gran cantidad de cheques se realiza en intervalos cortos de tiempo. La CCC, está normada mediante el reglamento de ley para las instituciones financieras, por esta razón, las entidades financieras están obligadas a remitir la información de los cheques pagados o rechazados, en un plazo máximo de 24 horas¹. El no cumplir con los plazos indicados en la ley y los reglamentos citados, puede provocar multas y sanciones a la institución financiera². Además, el no poder cobrar el monto de cheques diariamente, es una pérdida económica para la institución. El informe presentado por el BCE, en el año 2018

¹ Banco Central del Ecuador, «Regulación No. 046-2013» Quito, 2014

² Banco Central del Ecuador, «Normas Generales Para Las Instituciones Del Sistema Financiero, Artículos, 23, 28.» , 2014.

mostró que se realizaron un promedio de 127 mil operaciones basadas en cheques, los cuales pasaron por la CCC³. Por tal motivo es necesario contar con sistemas de verificación computarizados ágiles y de bajo coste.

Podemos definir el término “firma manuscrita” a un trazo o gráfico que una persona realiza a mano sobre un documento con el fin de dar validez y de expresar conformidad o aceptación al contenido de este documento [4, 5]. Cada persona desarrolla ciertos patrones en los trazos de la firma que permiten reconocer una firma auténtica de una falsificada.

En este trabajo de tesis se propone implementar, evaluar y comparar varios algoritmos supervisados para clasificación de firmas manuscritas, con el objeto de identificar al propietario de esta. Posteriormente se determinará el algoritmo de mejor rendimiento que permita verificar una firma en el menor tiempo. De esta manera se pretende reducir costos en tiempo y personal, con el objetivo de ser competitivo frente al proceso de verificación manual de la firma.

Los métodos de verificación o identificación de firmas manuscritas existentes en la actualidad se pueden clasificar en: verificación de firmas manuscritas fuera de línea (offline), y verificación de firmas manuscritas en línea (online) [6]. El método offline se refiere a analizar y procesar la imagen de la firma manuscrita después que el usuario ha terminado de firmar. Mientras que el reconocimiento en línea (online) es el análisis de la firma manuscrita cuando el usuario la escribe usando una tableta digitalizadora y un bolígrafo digitalizador. Esto tiene como ventaja la posibilidad de contar con la firma original y algunas características adicionales como presión, velocidad, punto de inicio y finalización de las líneas, entre otras [7].

Respecto al método offline, existen dos problemas principales a resolver. El primer problema consiste en que se pierde la información dinámica o secuencial de la firma, tales como la velocidad al dibujar la firma manuscrita por parte del firmante, la presión del lápiz sobre el papel [8]. Un segundo problema es que, en la práctica, es demasiado caro, demasiado difícil o incluso imposible obtener un número significativo de firmas de buena calidad por cada individuo en la base de datos [9].

³ Banco Central del Ecuador, « Informe rendición de cuentas 2018» Quito, 2019

1.2. PLANTEAMIENTO DEL PROBLEMA

En este proyecto de titulación usaremos específicamente técnicas offline. Para lograr nuestro objetivo, en primer lugar, se procederá a la recolección de firmas manuscritas de personas residentes en el Ecuador, luego se construirá una base de datos con la cual se realizará la validación y comparación de los algoritmos de clasificación supervisados.

En cuanto a los métodos de reconocimiento de firmas manuscritas, en base a la literatura se seleccionará tres algoritmos de clasificación supervisados adaptándolos e implementándolos en lenguaje Python, se realizará un pre-proceso de la firma (imagen), para la supresión de ruido y normalización de la imagen para obtener atributos más relevantes de la firma, posteriormente se obtendrán las características de cada firma manuscrita y se ponderará su importancia con el objeto de evaluar y comparar su rendimiento.

Se debe mencionar que este proyecto aún está lejos de proponer una solución para reemplazar una validación y autenticación de firma manuscrita por un experto humano, sin embargo, este trabajo puede ser una alternativa útil, para crear herramientas rápidas y de bajo costo para identificar firmas manuscritas que no se falsifican por expertos y que podría aplicarse cuando se trata de identificar firmas manuscritas en una gran cantidad de documentos.

1.3. OBJETIVO GENERAL

Implementar, evaluar y comparar algoritmos de clasificación supervisada relacionados con el tratamiento de imágenes de firmas manuscritas para identificar al propietario de una firma manuscrita.

1.4. OBJETIVOS ESPECÍFICOS

- Explorar diferentes algoritmos de clasificación supervisada orientados al procesamiento de imágenes de firmas manuscritas e identificar al menos tres de estos algoritmos, que se aproximen a nuestras necesidades.
- Adaptar los algoritmos seleccionados a nuestras necesidades específicas e implementarlos usando el lenguaje de programación Python y el banco de datos de firmas manuscritas obtenido localmente.

- Experimentar estos algoritmos para evaluarlos y compararlos de acuerdo a su rendimiento basado en el análisis de la tasa de error, de exactitud y precisión. Se espera alcanzar una tasa de error inferior al 8% en la fase de entrenamiento y como mínimo una tasa de error inferior al 12% en la fase de validación.

1.5. MARCO TEÓRICO

En esta sección se detallan algunas definiciones relacionadas con el tratamiento, procesamiento de imágenes y posterior clasificación de la firma.

1.5.1. Reconocimiento de la firma manuscrita como sistema biométrico

Podemos definir el término “firma manuscrita” a un trazo o gráfico que una persona realiza a mano sobre un documento con el fin de dar validez, de expresar conformidad o aceptación al contenido del mismo [4]. Cada persona desarrolla ciertos patrones en los trazos de la firma que permiten reconocer una firma autentica de una falsificada.



Figura 1. Ejemplo de una firma manuscrita.

Un sistema biométrico podría definirse como “un sistema automático que permite el reconocimiento de seres vivos a través de sus rasgos inherentes” [10]. La firma manuscrita es una modalidad biométrica que se utiliza con fines de reconocimiento e identificación de personas. La tecnología biométrica en la autenticación de firmas manuscritas se puede dividir en dos grandes métodos: métodos estáticos (offline), métodos dinámicos (online) [10]. En este trabajo nos enfocaremos solamente en los métodos estáticos (offline), estos métodos estáticos verifican las características de la firma que no varían en el tiempo y normalmente nos referimos a imágenes escaneadas en documentos en donde se plasma la firma manuscrita tales como documentos legales, cheques, papeletas de depósito, entre otros.

Un problema a resolver es que en los métodos estáticos (off-line), la firma manuscrita plasmada en una imagen gráfica puede ser falsificada por una persona con experiencia, mientras que en los métodos dinámicos (on-line) las características de la firma tales como: velocidad, aceleración, tiempo, presión, son muy difíciles de reproducir [10].

1.5.2. Pre procesamiento de una imagen digital

El objetivo del pre-procesamiento es obtener a partir de una imagen digital inicial, una imagen final que sea adecuada para un posterior procesamiento de la misma, en esta fase el pre-procesamiento puede mejorar algunas características de la imagen. Uno de los problemas más comunes en una imagen es el ruido. “El ruido se puede definir como la variación del nivel de color o gris que tiene un pixel por causas distintas a la luz reflejada de los objetos”, En el caso de firmas off line, el ruido generalmente es creado por el scanner mientras la imagen es digitalizada [11], es importante eliminar el ruido en la imagen de la firma manuscrita para su posterior procesamiento. En las siguientes imágenes se puede observar el ruido que puede tener una imagen, casi imperceptible a simple vista.

Firma manuscrita



Imagen de la firma y el ruido existente encontrado después de un proceso de ecualización.

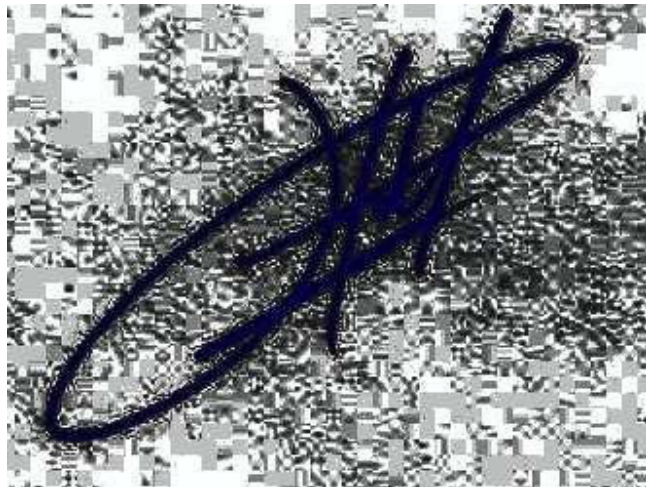


Figura 2. Ejemplo del ruido no visible de la imagen de una firma.

1.5.3. Histograma de una imagen.

Un histograma es una representación gráfica de la distribución de probabilidad de los distintos tonos de color o de gris de una imagen, dicho de otra manera, es una representación gráfica en donde se muestra la frecuencia con la que se repite cada nivel de color o de gris [12]. En el caso de imágenes en color se tendrán tres histogramas, uno por cada canal de color. A continuación, se muestran dos imágenes en color y en tonos de gris con sus histogramas.

Imagen en color Original
(2000x1500)



Histograma de la imagen en Color

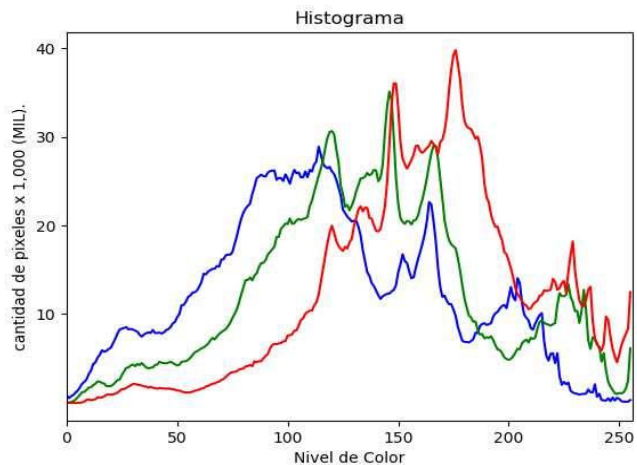


Figura 3. Histograma de una imagen en color.

Imagen original en tonos de Gris
(2000x1500)



Histograma de la imagen gris

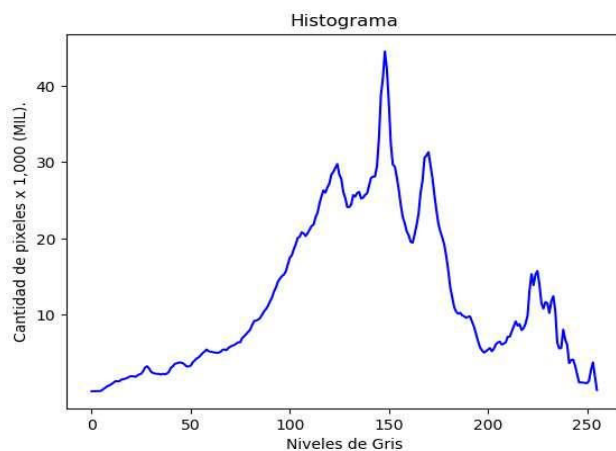


Figura 4. Histograma de una imagen en tonos de gris.

1.5.4. Ecuación del histograma de una imagen.

La ecualización de una imagen en base a su histograma es una transformación (función) cuyo propósito es lograr en lo posible tener una distribución uniforme para todo nivel de color o de gris, es decir que la imagen pueda tener un número de pixeles en lo posible similar para cada nivel de color o de gris [12]. Matemáticamente al aplicar esta operación, el histograma original de la imagen debe cambiar a un histograma con distribución uniforme sobre los niveles de color o de gris, esto no siempre sucede debido a que la variable independiente de esta transformación (tono de color o de gris) no es una variable continua, el tono de color o de gris es una variable discreta. Esta función de transformación está dada por la siguiente formula:

$$T(R_j) = \sum_{j=1}^k P_r(R_j) = \sum_{j=1}^k \frac{N_j}{N} \quad [13]$$

Ecuación 1. Ecuación de ecualización

Donde:

- k Es el número de niveles de gris
- Pr Es la probabilidad de un cierto nivel de gris
- Rj Es el nivel de gris j
- Nj Es el número de pixeles con un valor de gris j
- N Es el total de pixeles

La ecualización de una imagen tiene como objetivo acentuar el contraste visual, se considera que una imagen esta ecualizada si su histograma sigue una distribución de probabilidad en lo posible uniforme, es decir si todos los niveles de gris tengan una frecuencia similar [12].

Imagen gris original (150x300)



Histograma de la imagen en Gris

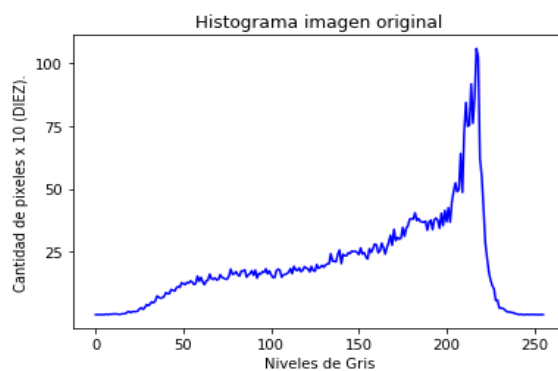


Figura 5. Imagen gris original para ser procesada usando ecualización

Imagen final ecualizada (150x300)



Histograma de la imagen ecualizada

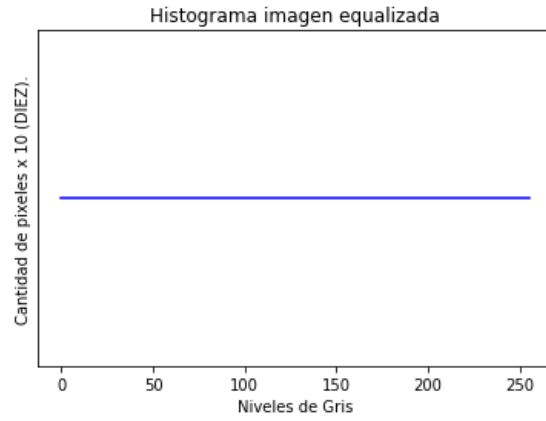


Figura 6. Imagen mejorada usando ecualización.

La figura 6, muestra una imagen intencionalmente escogida para mostrar el resultado óptimo al que se pretende llegar al someter la imagen al proceso de ecualización, como se puede observar el histograma de la misma sigue una distribución de probabilidad uniforme. En las figuras 7 y 8, se muestra una imagen escogida al azar con su histograma y después la misma imagen después de aplicar el método de ecualización al histograma de la imagen.

Imagen original (150x300)



Histograma de la imagen original

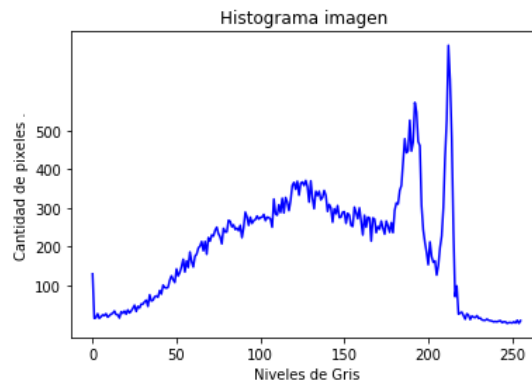


Figura 7. Imagen de un paisaje después del proceso de ecualización.

Imagen final ecualizada (150x300)

Histograma de la imagen ecualizada

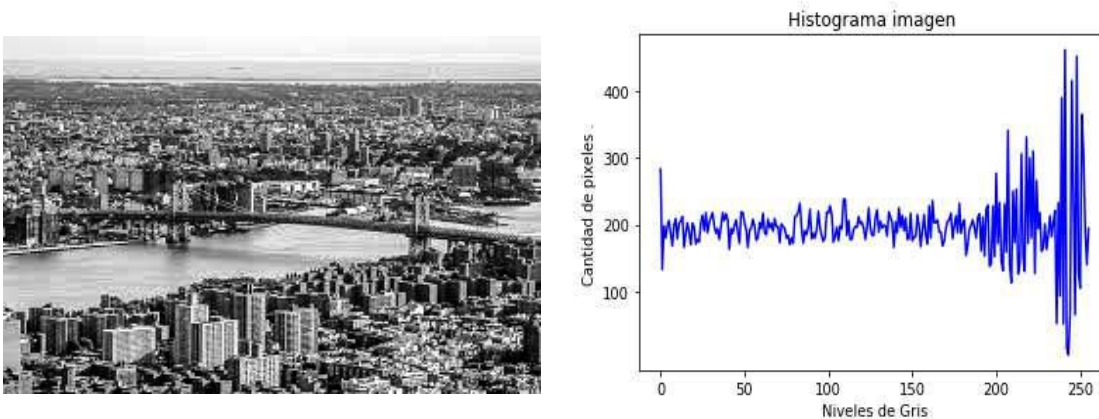
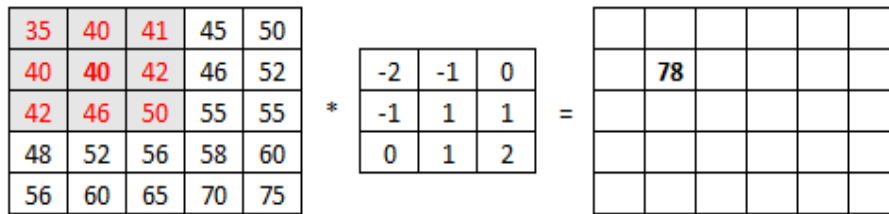


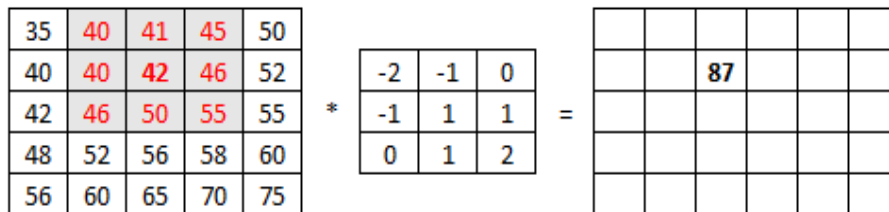
Figura 8. Imagen ecualizada y su histograma

1.5.5. Métodos de eliminación de ruido en imágenes

“El ruido es definido como cualquier perturbación que sufre una señal en el proceso de registro” [14]. Los métodos de eliminación de ruido se pueden clasificar en métodos espaciales y métodos basados en el dominio de la frecuencia, su uso depende del tipo de ruido (ruido gaussiano, ruido impulsivo, ruido repetitivo) que se encuentre en la imagen. Un método basado en el dominio de la frecuencia es el método de la Transformada de Fourier, que se basa en obtener el espectro de frecuencia para encontrar un tipo de ruido repetitivo [14]. Los métodos espaciales se basan en la aplicación de una operación de convolución de un kernel (matriz) con la imagen original, la idea de la operación de convolución se muestra en la figura 9. Los tipos básicos de filtrado espacial son: Suavizado, Media, Mediada, Bilateral, Filtrado de radio variable y Filtro Gaussiano [15], dependiendo del tipo de imagen se usa primero el método de ecualización del histograma, con el objetivo de magnificar el ruido y ser localizado más fácilmente en una imagen para luego aplicar un método de filtrado.



$$35*(-2)+40*(-1)+40*0+40*(-1)+40*1+42*1+42*0+46*1+50*2 = 78$$



$$40*(-2) + 41*(-1) + 45*0 + 40*(-1) + 42*1 + 46*1 + 46*0 + 50*1 + 55*2 = 87$$

Figura 9. Operación de convolución en una imagen gris.

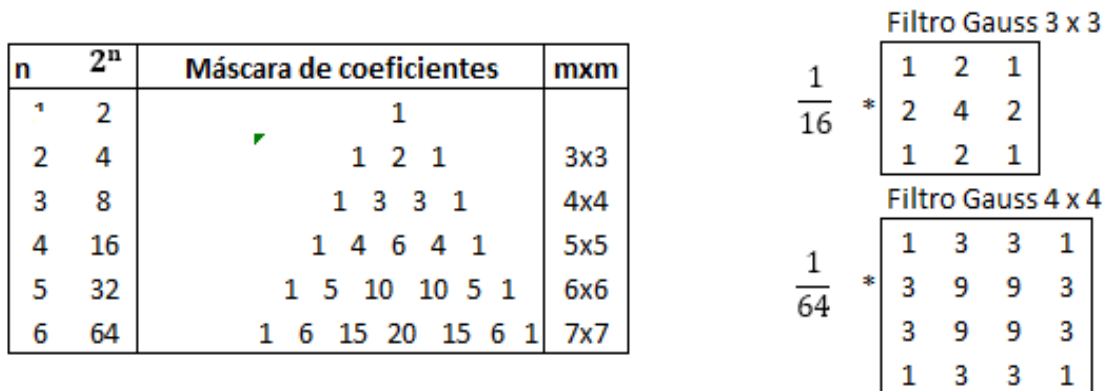


Figura 10. Triángulo de Pascal para generación de coeficientes gaussianos.

En la figura 10 se muestra la tabla denominada el Triángulo de Pascal que permite obtener la matriz (kernel) que se usa para hacer la convolución con la imagen original para la reducción de ruido.

Otro tipo de filtro que se usa para la eliminación del ruido es el de la media en donde cada pixel de salida tiene el promedio de brillo de sus vecinos. Se pueden cambiar los filtros para obtener distintos efectos.

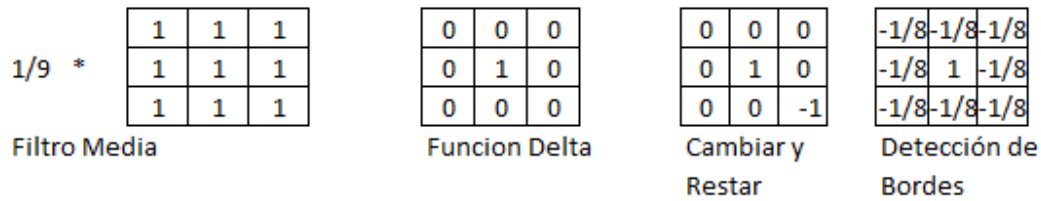


Figura 11. Ejemplo de filtros para usos diversos.

1.5.6. Esqueletización de imágenes en tonos de gris

La esqueletización de una imagen es un método que se usa para reconocer patrones. La idea de este método es eliminar píxeles de la imagen de una firma hasta que se obtiene la firma adelgazada con un grosor del tamaño de un pixel, este esqueleto tiene información importante de la forma y el patrón de una imagen [16]. Hay varios métodos publicados para encontrar el esqueleto de una imagen, pero no existe un método totalmente adecuado, debido a que en el proceso de esqueletizado las operaciones morfológicas usadas pueden introducir ruido indeseable en el esqueleto final de la imagen. En este trabajo se usará el algoritmo ZS que se basa en la teoría de los vecinos próximos examinando una imagen en regiones o matrices de 3x3 [16], en esta referencia se propone una modificación al algoritmo original propuesto por Zhang and Suen para encontrar el esqueleto de una imagen.

En el algoritmo de esqueletización propuesto en [16], se propone el método de ir quitando capa por capa de los píxeles de la frontera, aplicando operaciones morfológicas de manera repetida (operaciones como la erosión), teniendo mucho cuidado de no eliminar los píxeles que tienen la información de la forma de la imagen. El algoritmo propuesto en el trabajo de investigación [16], es el siguiente:

Notación usada para el algoritmo Zhang and Suen.

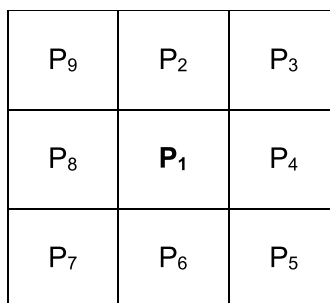


Figura 12. Notación usada en el algoritmo de esqueletización.

Prueba 1.- Un pixel del contorno es marcado para ser borrado si cumple las siguientes condiciones:

$$2 \leq B(P_1) \leq 6$$

$$A(P_1) = 1$$

$$P_2 * P_4 * P_6 = 0$$

$$P_4 * P_6 * P_8 = 0$$

Dónde: $B(P_1)$ es el número de vecinos con valor 0, de P_1 en una vecindad 8.
 $A(P_1)$, es el número de cambios de valor de 0 a 1 o de 1 a 0, alrededor de P_1 , siguiendo la secuencia $P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$. En la siguiente figura se tiene que $B(P) = 5$ y $A(P) = 4$.

0	1	0
0	P	0
0	1	1

Figura 13. Ejemplo de cambios de 0 a 1 y de 1 a 0.

Prueba 2.- Se marcan para eliminar los píxeles del contorno que cumplen las siguientes condiciones:

$$2 \leq B(P_1) \leq 6$$

$$A(P_1) = 1$$

$$P_2 * P_4 * P_8 = 0$$

$$P_2 * P_6 * P_8 = 0$$

El algoritmo se resume de la siguiente manera:

- a. Para todos los píxeles del contorno, se debe marcar como candidato a ser borrado si cumple las condiciones de la prueba 1.
- b. Una vez recorrido todos los puntos, se debe borrar o colocar el valor de cero (0) todos los píxeles marcados en el punto anterior.
- c. Para todos los píxeles del contorno, se debe marcar como candidato a ser borrado si cumple las condiciones de la prueba 2.
- d. Una vez recorrido todos los puntos, se debe borrar o colocar el valor de cero (0) todos los píxeles marcados en el punto anterior.
- e. Volver al paso a), mientras se sigan marcando píxeles.

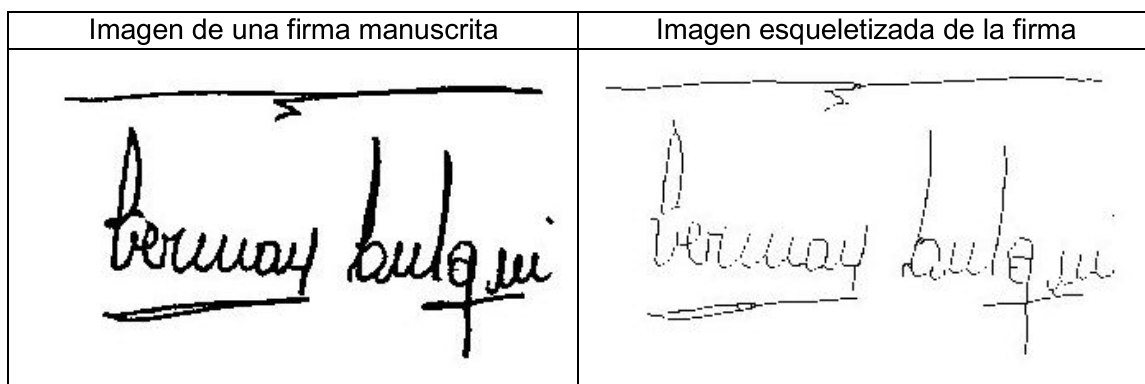


Figura 14. Imagen de una firma original y el esqueleto de la imagen.

1.5.7. Características de una imagen (descriptores)

En el proceso de clasificación de una imagen, es necesario establecer un grupo de características que permita definir una imagen, este grupo de características toman el nombre de descriptores, también se las conoce como vector de características. “Los descriptores son el primer paso para encontrar el vínculo entre los píxeles contenidos en una imagen digital y lo que representan” [17]. Del conjunto de descriptores o vector de características algunos representan características esenciales del objeto, las cuales no pueden ser eliminadas y otros representan características superficiales del mismo que podrían ser eliminadas, sin que ello implique confusión en la descripción del objeto. Antes de realizar la extracción de características de una imagen primero se debe reducir el efecto del ruido, para luego detectar regiones, determinan contornos de interés y finalmente refinar el resultado [17]. De manera

general las características o descriptores se pueden dividir como: descriptores globales, descriptores locales, descriptores geométricos, de contorno y por medio de regiones.

Los descriptores globales son características extraídas de toda la imagen, tales como: área, largo, ancho, relación del aspecto (aspect-ratio), orientación, centro de gravedad, máximo histograma horizontal (fila con máximo valor), máximo histograma vertical (columna con máximo valor), área de la imagen, proyección vertical y proyección horizontal [17].

Las características locales son descriptores que se extraen de una porción de la imagen, tales como: densidad de los píxeles, inclinación de los rasgos, puntos críticos [17].

Descriptores geométricos: Describen las características geométricas de la firma, tales como variaciones de la rotación, variación de estilo [17].

Descriptores de contorno: La forma geométrica de los contornos de los objetos debe ser analizada utilizando un esquema de representación. Dentro de los esquemas de representación podemos mencionar los códigos de cadena, los descriptores de Fourier, las formas, las aproximaciones poligonales, entre otros.

Descriptores por medio de regiones: Se utilizan cuando el interés principal es describir las propiedades de la región de los objetos detectados, como la intensidad, el color, la textura, etc. Existen varios descriptores que pueden ser calculados de forma muy sencilla sobre las regiones, algunos de ellos son el área, el número de Euler y la densidad, los descriptores geométricos que estudian la forma geométrica de los contornos de las regiones (objetos), tales como el alargamiento, la espesura, la dirección y la compacidad [17].

En este trabajo de investigación, la extracción de las características de la firma manuscrita no se realiza en forma separada del proceso de aprendizaje automático, sino que la extracción de características de la imagen de una firma se delega a la capa convolucional de una red neuronal convolucional, tema que se abordará en la sección de redes convolucionales, este concepto se conoce como Deep Learning.

1.5.8. Redes neuronales

Las redes neuronales artificiales, están inspiradas en las redes neuronales biológicas, pero con unas estructuras de conexión y funcionalidades diferentes, estas ANN (*Artificial Neuronal Network*) tienen tres características principales [18].

Auto-organización y adaptabilidad: utilizan algoritmos de aprendizaje adaptativo y de auto-organización ofreciendo mejores posibilidades de procesamiento robusto [18].

Procesado no lineal: aumenta la capacidad de la red para la clasificación de patrones, la aproximación de funciones, y el aumento de su inmunidad frente al ruido [18].

Procesado paralelo: normalmente se usa un gran número de nodos de procesamiento, con alto nivel de interconectividad [18].

El elemento básico de computación en una red neuronal se llama nodo, unidad o neurona, este elemento recibe datos de entrada desde la salida de otros nodos o del exterior. A cada entrada "X" se le asigna un peso específico, un valor asociado "W", el cual será determinado por aumento o disminución de su valor en el proceso de aprendizaje [18].

$$U = \sum_{i=1}^n X_i \cdot W_i + \theta$$

Ecuación 2. Producto de entradas y pesos más una constante.

Como segundo componente de una neurona tenemos una función f, llamada función de activación, que toma como dato de entrada la suma de las entradas X_i multiplicadas por los pesos W_i . Esta función de activación es una regla que determina el efecto que se aplicará a la entrada total U [19].

$$Y = f(\sum_{i=1}^n X_i \cdot W_i + \theta)$$

Ecuación 3. Estructura matemática de una neurona

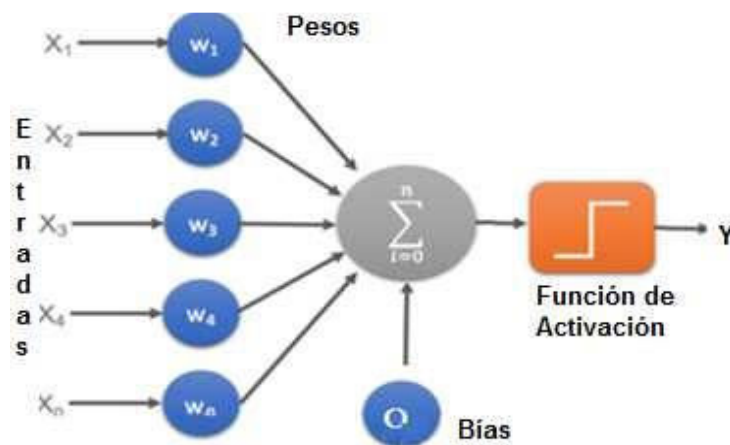


Figura 15. Diagrama de un nodo o neurona (ecuación 3)

Donde:

X_i : Son las señales de entrada cuyo valor provienen de otros nodos o de entradas externas.

W_i : Pesos asociados a cada entrada X_i .

f : Función no lineal simple llamada función de activación

θ : Constante, valor llamado sesgo (bias en inglés), cuyo objeto es evitar salidas o resultados no deseados.

Las funciones de activación de uso común son: función escalón, función lineal y mixta, función tangente hiperbólica y función sigmoidea [20].

1.5.8.1. Función escalón

La función de activación escalón se asocia a neuronas binarias en las cuales, cuando la suma de las entradas es mayor o igual que el umbral de la neurona, la activación es 1; si es menor, la activación es 0 ó -1.

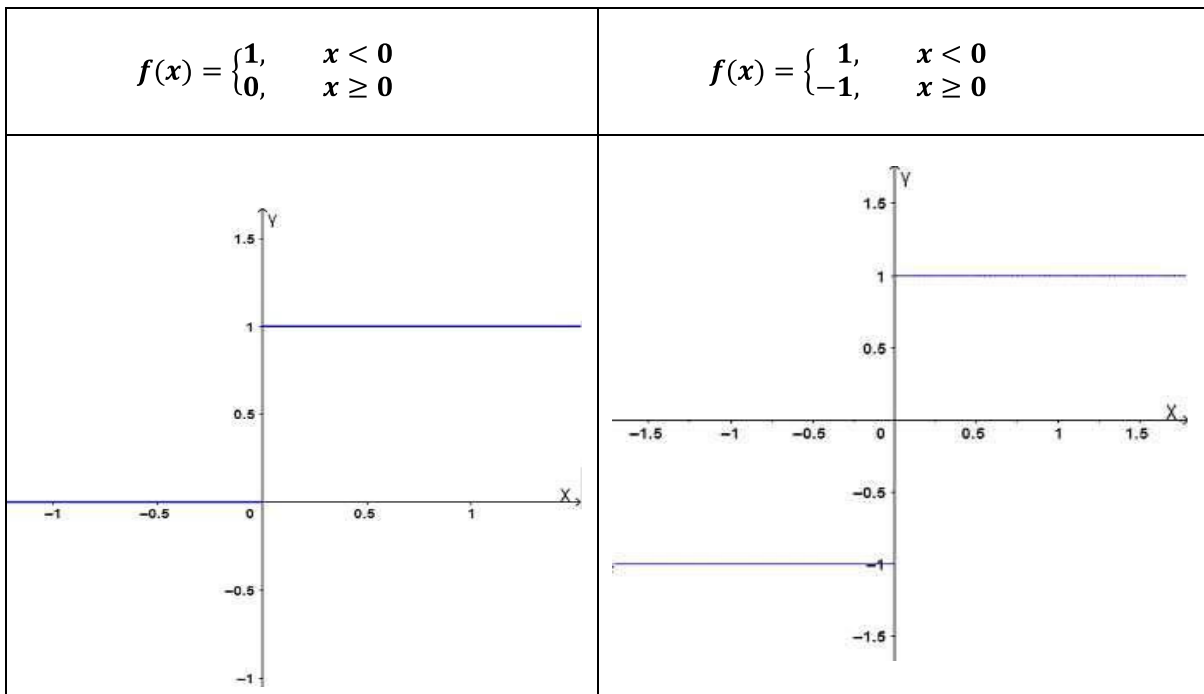


Figura 16. Función escalonada [20].

1.5.8.2. Funciones lineal y mixta

La función lineal está definida por $f(x) = x$, a esta función se la puede combinar con funciones que anulan los valores que toma U , por debajo de un límite inferior, devolviendo el valor de 0, y cuando el valor U , está por encima de un límite superior toma el valor de 1, si el valor de U esta entre estos dos límites entonces el resultado entregado es igual al de la función lineal. De manera general se puede definir un límite inferior y superior $-C$ y C respectivamente.

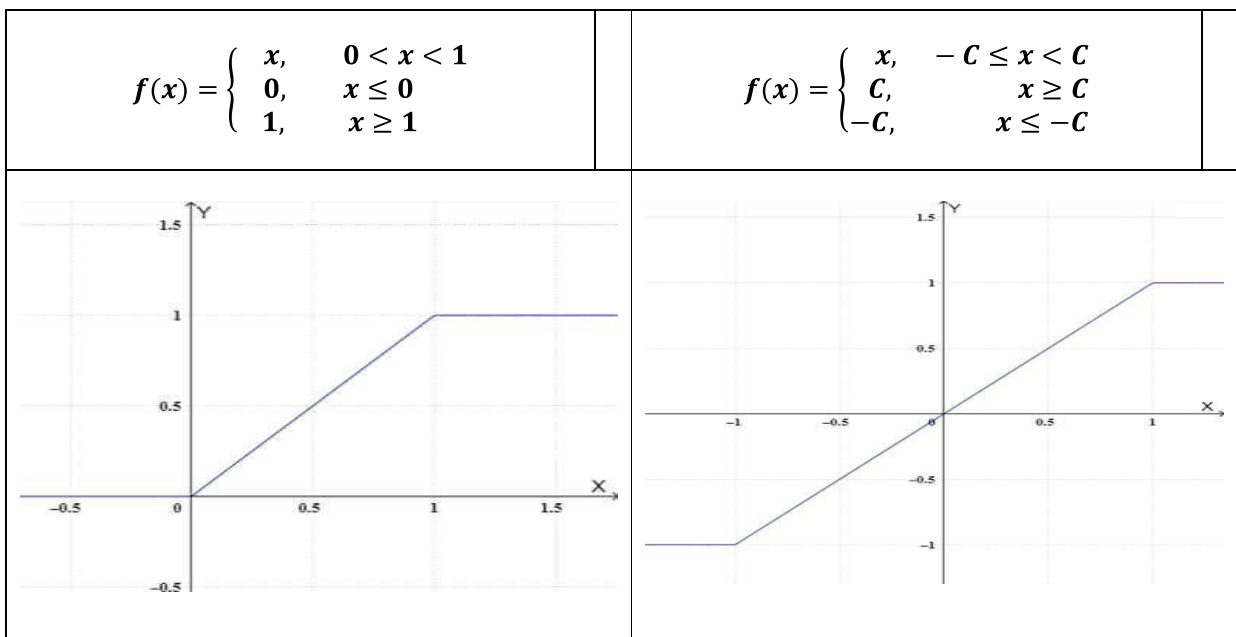


Figura 17. Funciones lineales mixtas [20].

1.5.8.3. Función tangente hiperbólica

La función de activación tangente hiperbólica se emplea en los casos que presentan variaciones suaves de valores positivos y negativos de la señal a clasificar. Esta función de activación es la más usada en un entrenamiento supervisado y de propagación por error [19].

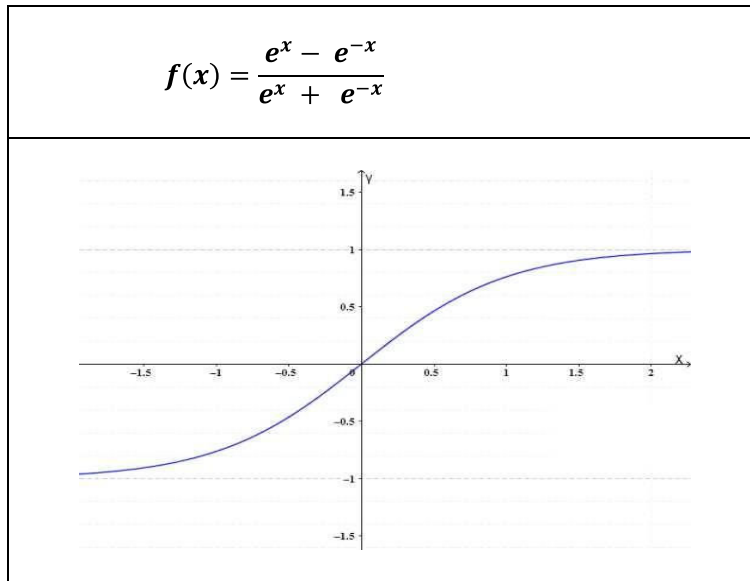


Figura 18. Función tangente hiperbólica [20].

1.5.8.4. Función sigmoidea

La importancia de la función Sigmoidea es que su derivada siempre es positiva y cercana a cero para los valores grandes positivos o negativos.

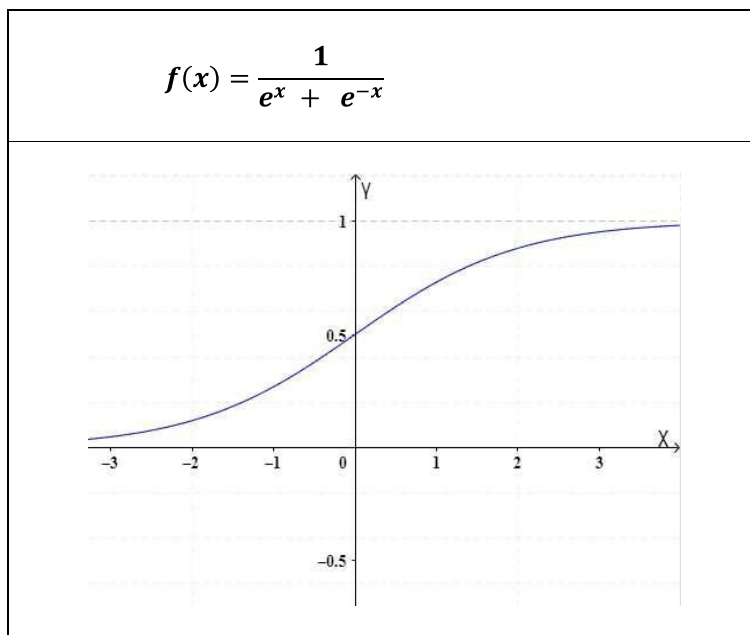


Figura 19. Función sigmoideal [20].

1.5.8.5. Redes neuronales convolucionales

Una red neuronal convolucional es un tipo de red neuronal, que lleva su nombre por el concepto matemático de “convolución” (definido por la ecuación 4), que es una transformación lineal de dos funciones en una tercera que puede representar una magnitud de valor real, en la que se superponen las anteriores [8]. En este trabajo se ha escogido utilizar este tipo de redes debido a que son extremadamente eficientes en el reconocimiento de objetos en imágenes.

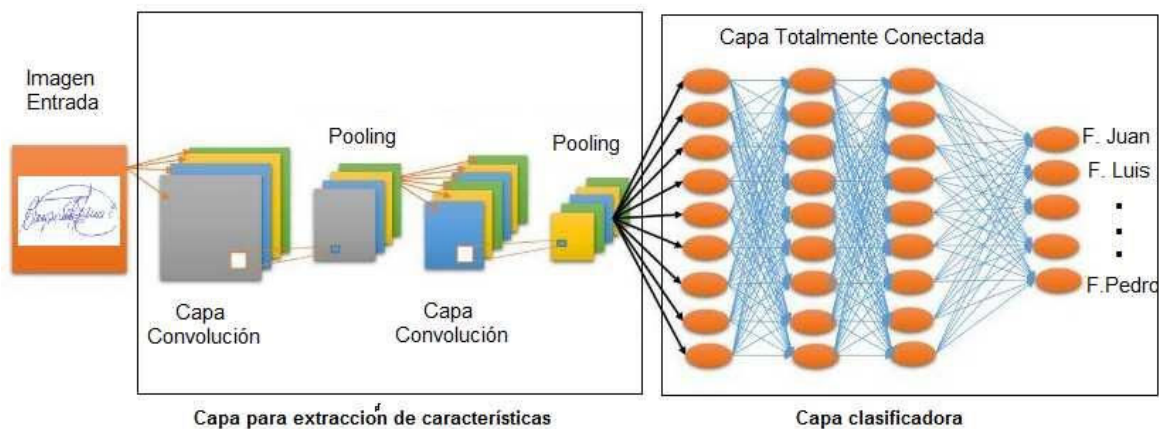


Figura 20. Arquitectura de una red neuronal convolucional

Este tipo de red está formada básicamente por dos grandes capas: la capa de convolución y la capa clasificadora. La primera capa se encarga de realizar la operación de extracción de las características de la imagen, mientras que el objetivo de la segunda capa es realizar la clasificación tomando como entrada las características extraídas en la primera etapa [21]. La primera capa, tiene a su vez una estructura conformada por varias capas de convolución y capas de reducción (pooling), en donde se pueden realizar operaciones de relleno, agrupación apilado y de avances o zancadas. En el procesamiento de imágenes, este tipo de arquitectura produce mejores resultados que los modelos de redes neuronales tradicionales que son totalmente conectadas, con un número menor de neuronas entrenadas [8].

Definición.- Dada una matriz A de dimensiones $m \times n$, y una matriz C de dimensiones $(2n+1) \times (2n+1)$, en donde: $2n+1 < m$ y $2n+1 < n$, se define la convolución de estas matrices como una nueva matriz $D = A * C$, definida a partir de la siguiente expresión [22].

$$D(i, j) = \frac{1}{k} \sum_{r=1}^{2N+1} \sum_{s=1}^{2N+2} A(i - N + r - 1, i - N + s - 1) * C(r, s)$$

Donde, $k = \sum_{i=1, j=1}^{2N+1} C(i, j)$

Ecuación 4. Operación de convolución de matrices

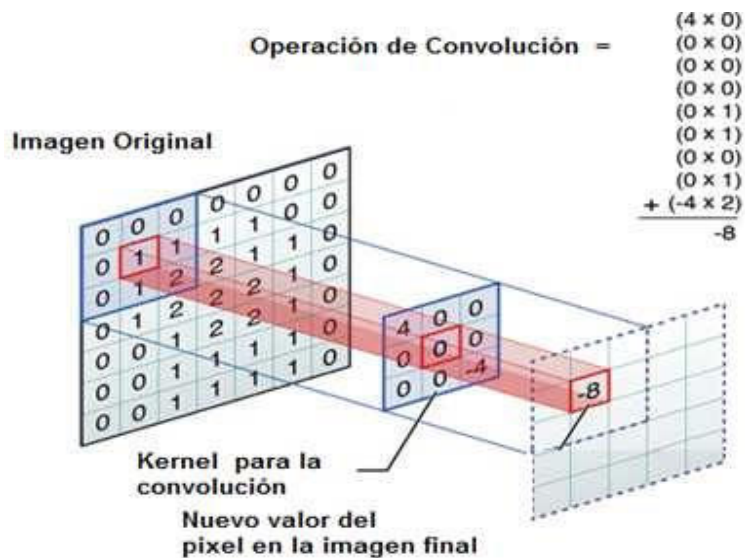


Figura 21. Operación de convolución en imágenes de tonos de gris.

Relleno (Padding). - En la capa de convolución se procesa la imagen usando filtros (matrices), la imagen resultante producto de esta convolución es de una dimensión menor a la dimensión de la imagen original, tal como se muestra en la figura 22.

IMAGEN	FILTRO	RESULTADO CONV.																																																													
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>0</td><td>1</td><td>2</td><td>7</td><td>4</td></tr> <tr><td>1</td><td>5</td><td>8</td><td>9</td><td>3</td><td>1</td></tr> <tr><td>2</td><td>7</td><td>2</td><td>5</td><td>1</td><td>3</td></tr> <tr><td>0</td><td>1</td><td>3</td><td>1</td><td>7</td><td>8</td></tr> <tr><td>4</td><td>2</td><td>1</td><td>6</td><td>2</td><td>8</td></tr> <tr><td>2</td><td>4</td><td>5</td><td>2</td><td>3</td><td>9</td></tr> </table>	3	0	1	2	7	4	1	5	8	9	3	1	2	7	2	5	1	3	0	1	3	1	7	8	4	2	1	6	2	8	2	4	5	2	3	9	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	1	0	-1	1	0	-1	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>5</td><td>4</td><td>0</td><td>-8</td></tr> <tr><td>10</td><td>2</td><td>-2</td><td>3</td></tr> <tr><td>0</td><td>2</td><td>4</td><td>7</td></tr> <tr><td>3</td><td>2</td><td>3</td><td>16</td></tr> </table>	5	4	0	-8	10	2	-2	3	0	2	4	7	3	2	3	16
3	0	1	2	7	4																																																										
1	5	8	9	3	1																																																										
2	7	2	5	1	3																																																										
0	1	3	1	7	8																																																										
4	2	1	6	2	8																																																										
2	4	5	2	3	9																																																										
1	0	-1																																																													
1	0	-1																																																													
1	0	-1																																																													
5	4	0	-8																																																												
10	2	-2	3																																																												
0	2	4	7																																																												
3	2	3	16																																																												
6x6	3x3	4x4																																																													

Figura 22. Imagen resultante sin relleno (padding)

Para evitar este efecto e la convolución se procede a rellenar la imagen original (figura 23, izquierda), con un valor predeterminado con el objeto de que al aplicar el filtro y realizar la operación de convolución, con el objetivo de que la imagen resultante tenga la misma dimensión que la imagen original (figura 23, derecha).

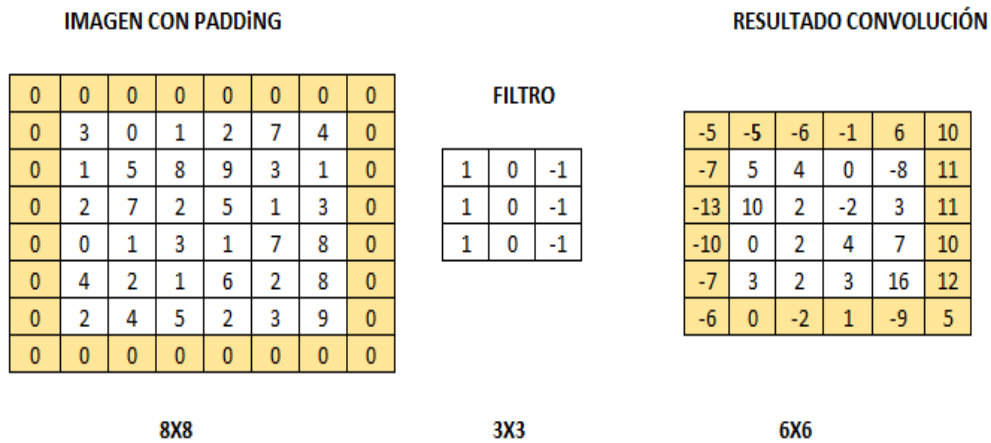


Figura 23. Operación de relleno (padding)

Agrupación (Pooling).- Es una operación que permite analizar el contenido de una imagen por regiones o por bloques, para extraer la información más representativa o menos representativa dependiendo de la aplicación que se desee realizar. El pooling permite reducir la cantidad de datos entre una capa y otra reduciendo el tiempo de procesamiento de entrenamiento de la red y preservando la información más o menos relevante.

La forma de reducir la imagen se realiza mediante la extracción de valores estadísticos como el promedio, el máximo o el mínimo de una región fija de la imagen original.

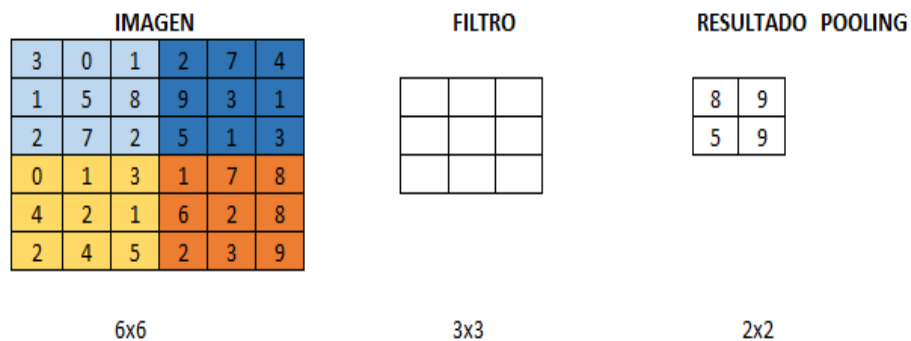


Figura 24. Operación de pooling en un área de 3x3, valor máximo

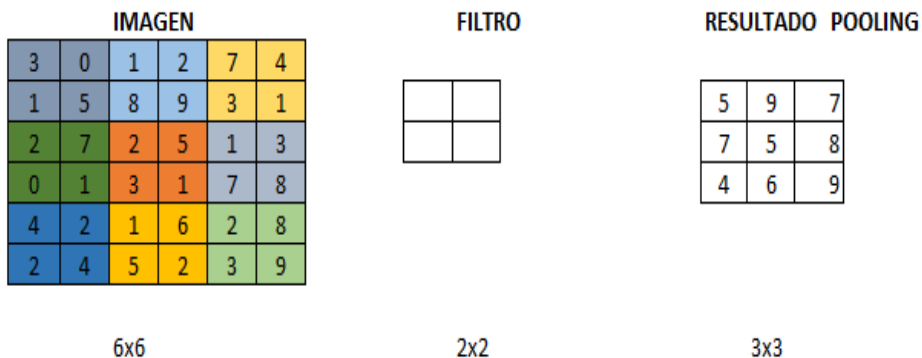


Figura 25. Operación de pooling en un área de 2x2, valor máximo

Apilado (Stacking).- Es el resultado de apilar múltiples imágenes producto de la convolución con múltiples filtros; en la siguiente figura se muestra una imagen de una firma a la cual se le ha aplicado cuatro filtros (kernel) distintos, un filtro a 0°, 45°, 90° y 135°, dando como resultado cuatro imágenes apiladas que contienen características distintas de la misma imagen.

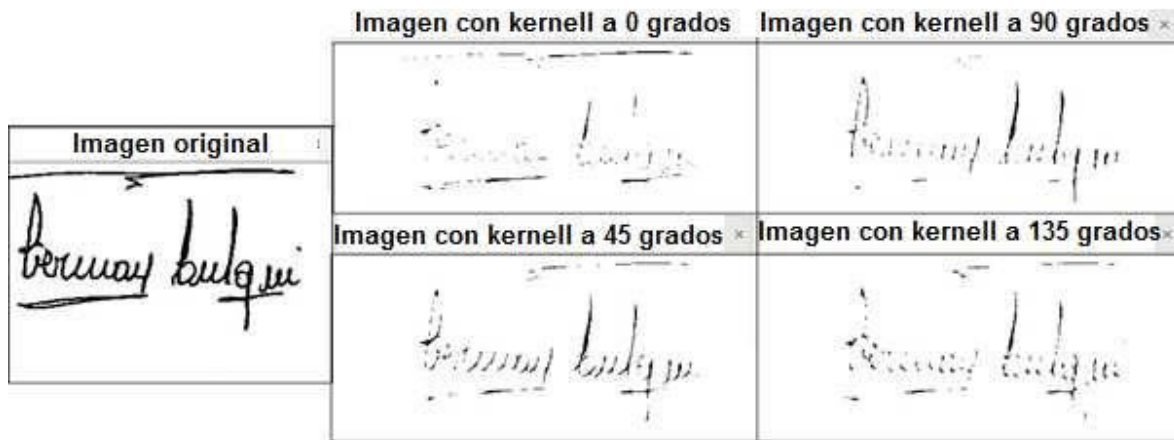


Figura 26. Ejemplo de stacking de una imagen.

Avances (Strides).- Esta operación hace referencia a la cantidad de pixeles que salta o avanza el filtro al recorrer la imagen original, en la operación de convolución.

IMAGEN						FILTRO						
3	0	1	2	7	4	1	0	-1	5	4	0	-8
1	5	8	9	3	1	1	0	-1	10	2	-2	3
2	7	2	5	1	3	1	0	-1	0	2	4	7
0	1	3	1	7	8							
4	2	1	6	2	8							
2	4	5	2	3	9							

Figura 27. Ejemplo de imagen con stride = 1

IMAGEN						FILTRO				
3	0	1	2	7	4	1	0	-1	5	4
1	5	8	9	3	1	1	0	-1	0	2
2	7	2	5	1	3	1	0	-1		
0	1	3	1	7	8					
4	2	1	6	2	8					
2	4	5	2	3	9					

Figura 28. Ejemplo imagen con stride = 2

1.5.9. Reconocimiento de patrones y clasificación

Un patrón se puede definir como un conjunto de características (descriptores) que permiten definir una imagen, un conjunto de patrones similares forman una clase. En el procesamiento de imágenes de firmas manuscritas, el proceso de verificar, autenticar o encontrar al dueño de la misma, se reduce a realizar un reconocimiento de patrones, para luego asociar cada patrón a la clase a la cual pertenece [15]. El proceso en el cual los descriptores son categorizados se conoce como clasificación y el algoritmo que implementa esta tarea se llama clasificador [17]. Los problemas básicos que se presentan en el proceso de clasificación son: a) conocer cuáles son los patrones más adecuados para representar las imágenes de firmas manuscritas y b) que número de patrones se debe usar.

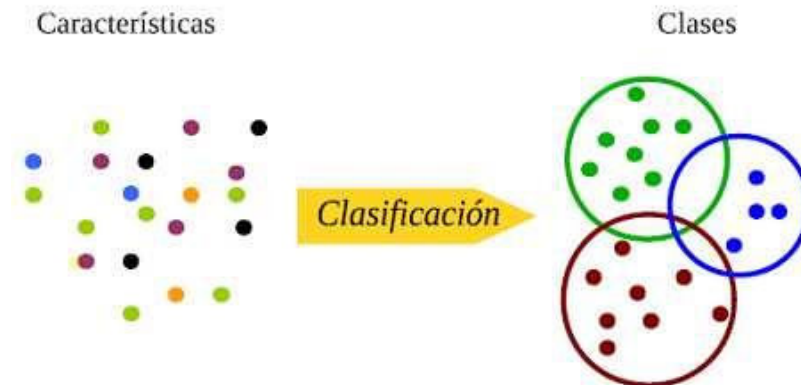


Figura 29. Clasificación en base a las características [15].

Para resolver el problema de clasificación una de las técnicas que se aplica es “algoritmos de aprendizaje automático” conocido como Machine Learning.

Los algoritmos de aprendizaje automático o Machine Learning (ML) se dividen en: no supervisados, supervisados y algoritmos híbridos que son la combinación de los dos anteriores [23]. Dentro de los algoritmos de clasificación supervisada se encuentran las Redes Neuronales Convolucionales (CNN).

El reconocimiento supervisado se utiliza cuando se tiene información a priori al proceso de reconocimiento. También es conocido como clasificación con aprendizaje, debido a que ciertas clases se encuentran ya definidas [15]. Dentro del reconocimiento supervisado podemos encontrar una división de acuerdo con el tipo de clasificador que se use, se pueden mencionar los siguientes:

Clasificadores Bayesianos: Estos clasificadores tienen su base en la teoría de la probabilidad y la estadística, utilizan análisis de varianzas, covarianzas, dispersión y distribución, principalmente. Se representan por medio de una red bayesiana, que es un grafo a cíclico dirigido en el que cada nodo representa una variable y cada conexión una dependencia probabilística [15].

Clasificadores Lineales: Las funciones discriminantes que rigen este tipo de clasificadores son combinaciones lineales que permiten separar los componentes del grupo de características [15].

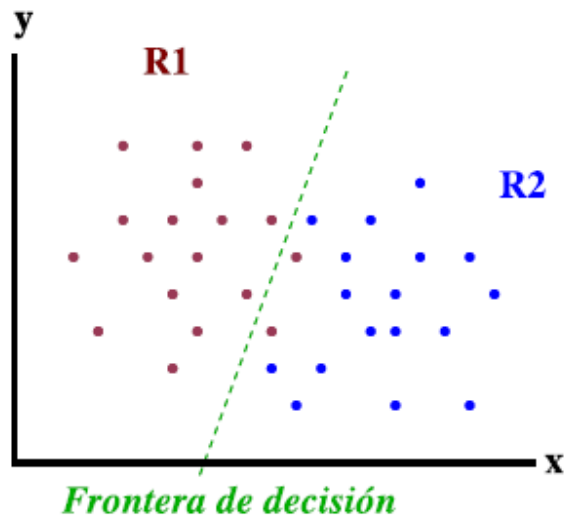


Figura 30. Clasificador lineal [15]

Clasificadores No lineales: Los bordes de decisión son expresados con funciones cuadráticas, las que pueden ser cónicas: parábolas, círculos, elipses, hipérbolas, de Fourier, entre otras. Las redes neuronales convolucionales se catalogan como clasificadores no lineales [15].

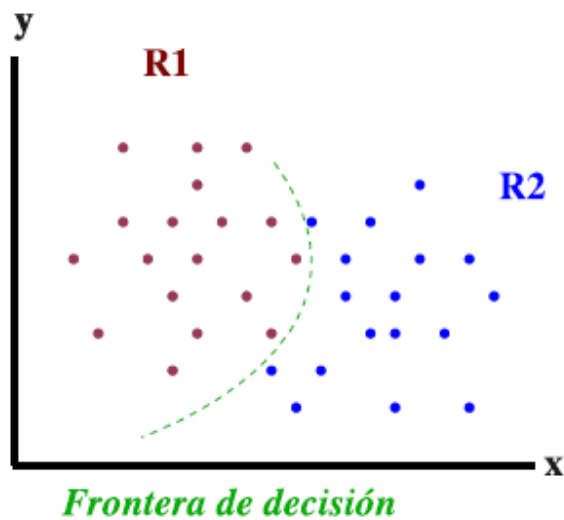


Figura 31. Clasificadores no lineales. [15]

1.5.10. Métricas de Evaluación

Para evaluar el rendimiento de un modelo de red neuronal se usa métricas de evaluación que toman como base valores de la fase de entrenamiento y de la fase de prueba, tales como: el error medio cuadrático (EMC), el cual mide el valor promedio de los errores elevados al cuadrado, la exactitud (accuracy) que nos muestra lo cercano que el valor predicho se encuentra del valor real, la función de pérdida (LOSS), que también se la conoce como función de costo y nos informa sobre qué tan efectivo es el modelo, la matriz de confusión la cual relaciona y compara los valores reales con los valores que fueron predichos en la ejecución del algoritmo [24].

Matriz de Confusión		Valores Reales	
		Positivo	Negativo
Valores Predichos	Positivo	VP	FP
	Negativo	FN	VN

Figura 32. Matriz de confusión [25].

De la Matriz de Confusión se obtiene los indicadores que se describen en la tabla 1, estos se pueden usar para medir la eficacia de una red neuronal.

Métrica	Descripción	Fórmula
Exactitud	Con que frecuencia es correcto el clasificador, es la relación entre el número total de predicciones correctas y el número total de predicciones. Conocida también por su nombre en inglés, como Accuracy .	$\frac{VP + VN}{VP + VN + FP + FN}$
Error de clasificación	Con que frecuencia es incorrecto el clasificador.	$\frac{FP + FN}{VP + VN + FP + FN}$
Sensibilidad	Cuando el valor real es positivo, con qué frecuencia es correcta esta predicción. Se conoce en inglés como Recall , también es conocida como tasa verdadera positiva, muestra que tan sensible es el clasificador para encontrar instancias positivas. Es la relación entre el número total de predicciones positivas correctas y el número total de predicciones positivas.	$\frac{VP}{FN + VP}$
Especificidad	Cuando el valor real es negativo, con qué frecuencia es correcta la predicción.	$\frac{VN}{FP + VN}$
Tasa de falsos positivos	Cuando el valor real es negativo, con qué frecuencia la predicción es incorrecta.	$\frac{FP}{FP + VN}$
Precisión	Si se predice un valor positivo, con qué frecuencia esta predicción es correcta. Es la relación entre las predicciones correctas y el total de predicciones previstas como correctas. Se debe enfatizar que la precisión no es lo mismo que la exactitud.	$\frac{VP}{VP + FP}$
F1	Promedio ponderado de precisión y sensibilidad, se la conoce como la medida armónica de la memoria y la precisión, mientras más alto es este valor, se considera que mejor es el modelo.	$\frac{2 * \text{Precision} \times \text{Sensibilidad}}{\text{Precision} + \text{Sensibilidad}}$

Tabla 1. Métricas de evaluación para las redes neuronales [26].

Donde:

VP (Verdadero Positivo), se predijo positivo y es cierto. Es decir que la firma perteneciente a una persona A, la red neuronal predijo que dicha firma pertenece a esta persona A.

VN (Verdadero Negativo), se predijo negativo y es cierto. Es decir, se predijo que la firma no pertenecía a una persona X, y la firma no pertenece a esa persona X.

FP (Falso Positivo), se predijo positivo, pero es falso. La red neuronal predijo que la firma pertenece a una persona X, pero la firma no pertenece a esa persona X. Error tipo 1.

FN (Falso Negativo), se predijo negativo, pero es falso. Es decir que se predijo que la firma no pertenece a una persona X, pero la firma si es de esa persona X. Error tipo 2.

La matriz de confusión mostrada en la figura 32, es la matriz que se usa en el caso de tener una clasificación binaria es decir en el caso de que existan solamente dos clases, para el caso que se tanga una clasificación de más de dos clases (multiclase), el análisis de la matriz de confusión difiere. A continuación se muestra una matriz de confusión multiclase en donde las letras A, B,..., Xn, indican las n clases existentes, Los valores VPi, indican los valores verdaderos positivos, y los valores nombrados como Ekj (donde k es un índice que no cambia) son los valores falsos positivos, los valores nombrados como Eik (donde k es un índice que no cambia) son los valores falsos negativos.

MATRIZ CONFUSIÓN		VALORES REALES											
		A	B	C	D	E	F	G	H	.	.	.	Xn
P R E D I C I Ó N	A	VP_A	EAB	EAC	EAD	EAE	EAF	EAG	EAH	.	.	.	EAXn
	B	EBA	VP_B	EBC	EBD	EBE	EBF	EBG	EBH	.	.	.	EBXn
	C	ECA	ECB	VP_C	ECD	ECE	ECF	ECG	ECH	.	.	.	EcXn
	D	EDA	EDB	EDC	VP_D	EDE	EDF	EDG	EDH	.	.	.	EDXn
	E	EEA	EEB	EEC	EED	VP_E	EEF	EEG	EEH	.	.	.	EEXn
	F	EFA	EFB	EFC	EFD	EFE	VP_F	EFG	EFH	.	.	.	EFXn
	G	EGA	EGB	EGC	EGD	EGE	EGF	VP_G	EGH	.	.	.	EGXn
	H	EHA	EHB	EHC	EHD	EHE	EHF	EHG	VP_H	.	.	.	EHXn
	Xn	EXnA	EXnB	EXnC	EXnD	EXnE	EXnF	EXnG	EXnH	.	.	.	VP_{Xn}

Tabla 2. Matriz de confusión multiclase [26].

Los valores de VP, VN, FP y FN se calculan por cada clase K. En la tabla siguiente el valor de C (i, j), es el valor de la celda en la posición (i, j), de la matriz en la tabla 2, donde i y j toman los valores desde A hasta Xn, donde n es el número de clases existentes.

Descripción	Valor para una clase determinada K
Verdaderos positivos (VP)	$VPk = \text{valor de la celda } C(k, k).$
Falsos Negativos (FN)	$FNk = \sum_{i=A}^{Xn} Eik, \text{ donde } i \text{ se mueve verticalmente.}$
Falsos Positivos (FP)	$FPk = \sum_{i=A}^{Xn} Eki, \text{ donde } i \text{ se mueve horizontalmente.}$
Verdaderos Negativos (VN)	$VNk = \sum_{i=A}^{Xn} \cdot \sum_{j=A}^{Xn} Cij; \text{ Donde } i \neq k \text{ y } j \neq k$

Tabla 3. Matriz de confusión multiclase, definición de valores VP, VN, FP y FN.

En las redes neuronales, en la etapa de entrenamiento se usan como indicadores de rendimiento del modelo: la exactitud y el error medio cuadrado (EMC), para medir la clasificación de los elementos que se predijeron como verdaderos, en relación a los que realmente son verdaderos.

La diferencia entre el valor real y el valor predicho se conoce como error de la predicción, el error medio cuadrático (EMC) se calcula como el promedio de los errores en las predicciones, elevado al cuadrado; mientras que la exactitud es la razón entre todos los elementos bien clasificados y el total de los datos. Para que la clasificación se considere óptima el valor de exactitud debe acercarse a 1, mientras que el valor EMC debe acercarse a 0.

$$EMC = \frac{1}{n} \sum_{i=1}^n (Pi - Ri)^2$$

Ecuación 5. Error medio cuadrático

Donde:

Pi es el valor predicho,

Ri es el valor real observado.

La métrica de evaluación exactitud (accuracy) no necesariamente es la mejor métrica o no siempre es la más adecuada a tomar en cuenta, en algunos casos puede resultar engañosa. Por esta razón se debe complementar la métrica de exactitud con otras como la precisión, la sensibilidad (Recall) y F1, esta última es una forma de combinar las métricas de sensibilidad y precisión.

En nuestro proyecto de investigación, se busca que la firma de una persona sea clasificada correctamente y que esta decisión sea la más confiable posible, por lo que estamos buscando la máxima precisión y máxima sensibilidad, aunque el algoritmo no sea cien por ciento exacto. Es decir que, en el presente trabajo, la métrica de precisión y sensibilidad deben complementarse con la métrica de exactitud (accuracy). Otra métrica importante para tomar en cuenta es la tasa de falsos positivos, es necesario que esta medida sea lo más baja posible, es decir que los casos en donde las predicciones indiquen que una firma pertenece a una persona y la firma no es de esta persona, en lo posible sea cero o sea un valor muy bajo (cerca de cero).

2. REVISIÓN DE LITERATURA.

Las fases que se siguieron para la revisión de literatura son las siguientes:

- a) Definir preguntas de investigación.
- b) Plantear términos de la búsqueda literaria (estrategia de búsqueda),
- c) Definición de bases de datos y motores de búsqueda.
- d) Establecer criterios para evaluación de calidad y seleccionar estudios basados en criterios de inclusión y exclusión.
- e) Búsqueda de documentos y clasificación de las investigaciones.
- f) Extraer y realizar la síntesis de datos.

2.1. PREGUNTAS DE INVESTIGACIÓN

Se han determinado tres preguntas de investigación que ayudarán en la revisión de la literatura de este proyecto, las cuales se mencionan a continuación.

La pregunta principal que se define en esta investigación es:

MQ: ¿Qué algoritmos de clasificación supervisada se han usado en la identificación de firmas manuscritas offline?

Las preguntas secundarias usadas para responder MQ se definen a continuación.

SQ1. ¿Cuáles son los métodos usados para el pre procesamiento de imágenes de firmas offline?

SQ2. ¿Cuáles son los algoritmos de clasificación supervisada para imágenes?

SQ3. ¿Qué algoritmos basados en redes neuronales profundas existen para clasificación supervisada de imágenes?

2.2. ESTRATEGIA DE BÚSQUEDA.

La revisión de literatura cubre el período 2011 a 2020. Inicialmente el rango de búsqueda empezaba desde el año 2000, pero se encontró que las soluciones planteadas en la primera década no involucraban avances en Machine Learning, a partir del año 2010, 2011 se encontró un incremento de trabajos en esta área. Las cadenas de búsqueda se concentrarán

en resolver los siguientes temas: clasificación de imágenes de firmas, redes neuronales y métodos de verificación de firmas offline y son:

“methods used for signatures image preprocessing.”.

“algorithms for supervised image classification”.

“neural network for supervised image classification”.

Estas cadenas fueron usadas para buscar en todo el cuerpo del documento, adicionalmente todas estas cadenas de búsqueda incluyeron la cadena “offline handwritten signature” en el título del documento, con el objeto de filtrar trabajos de investigación y obtener una lista de documentos corta y enfocada en el tema.

2.3. DEFINICION DE BASES DE DATOS Y MOTORES DE BÚSQUEDA.

Las fuentes de información a las que se accedió son IEEEExplore, Springer, ACM Library and Science Direct.

2.3.1. Definición de criterios de inclusión y exclusión

- Criterios de Inclusión:
 - ✓ Publicaciones científicas realizadas en revistas y conferencias (artículos) cuya área sea Computer Science.
 - ✓ Artículo científicos que tengan más de 20 citas.
 - ✓ Artículos científicos publicados en el período de 2011 al 2020.
 - ✓ Artículos científicos que presenten modelos, algoritmos, herramientas para la verificación offline de firmas.
 - ✓ Artículos que proponen métricas para la medición de la precisión en la verificación y clasificación offline de firmas.
 - ✓ Artículos que usen preferentemente redes neuronales.
- Criterios de Exclusión:
 - ✓ Artículos que no proponen o no aportan a la automatización en la clasificación offline de firmas.
 - ✓ Trabajos y tesis de pregrado.
 - ✓ Artículos científicos que hayan sido publicados antes del año 2011.

- ✓ Artículos que usen métodos de verificación de firmas distintas a offline.

2.3.2. Búsqueda y clasificación de estudios relevantes.

La información de los documentos encontrados fue sometida a filtros por fecha de publicación mayor al año 2011 y por los criterios de inclusión y exclusión indicados en la sección anterior, posteriormente fueron agrupados por base de datos y por el año de publicación, en la tabla 4 y figura 33, se muestra un resumen de los trabajos encontrados agrupados por base de datos y en la figura 34 se muestra los documentos agrupados por año de publicación y por base de datos, después de aplicar los criterios de inclusión y exclusión.

Preguntas	Bases de datos			
	Springer	Science Direct	IEEE	ACM
SQ1	7	4	24	2
SQ2	34	29	4	9
SQ3	26	13	33	10
TOTAL	67	46	61	21

Tabla 4. Resultados después de aplicar los criterios de inclusión y exclusión.

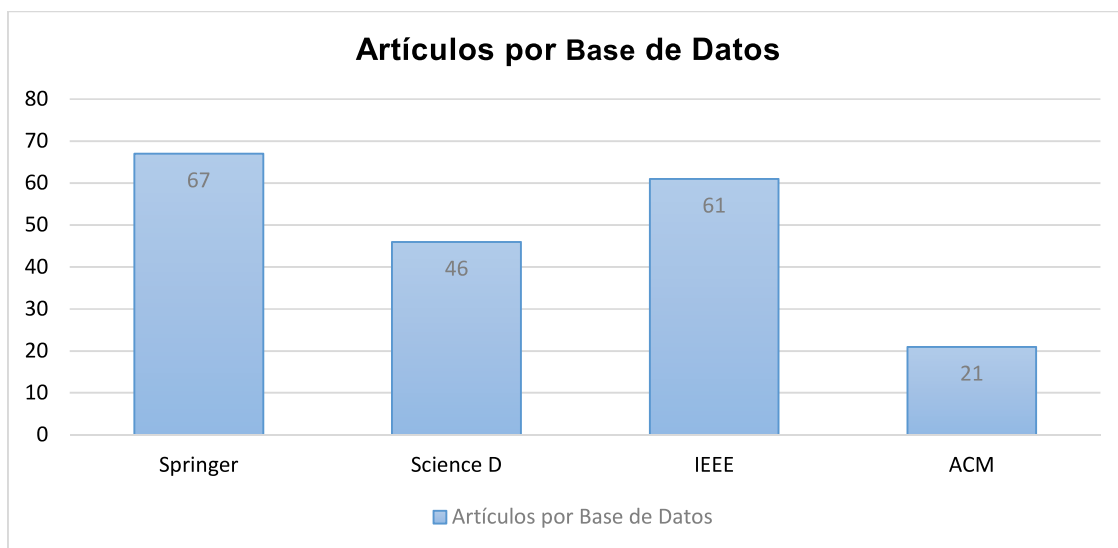


Figura 33. Publicaciones por base de datos

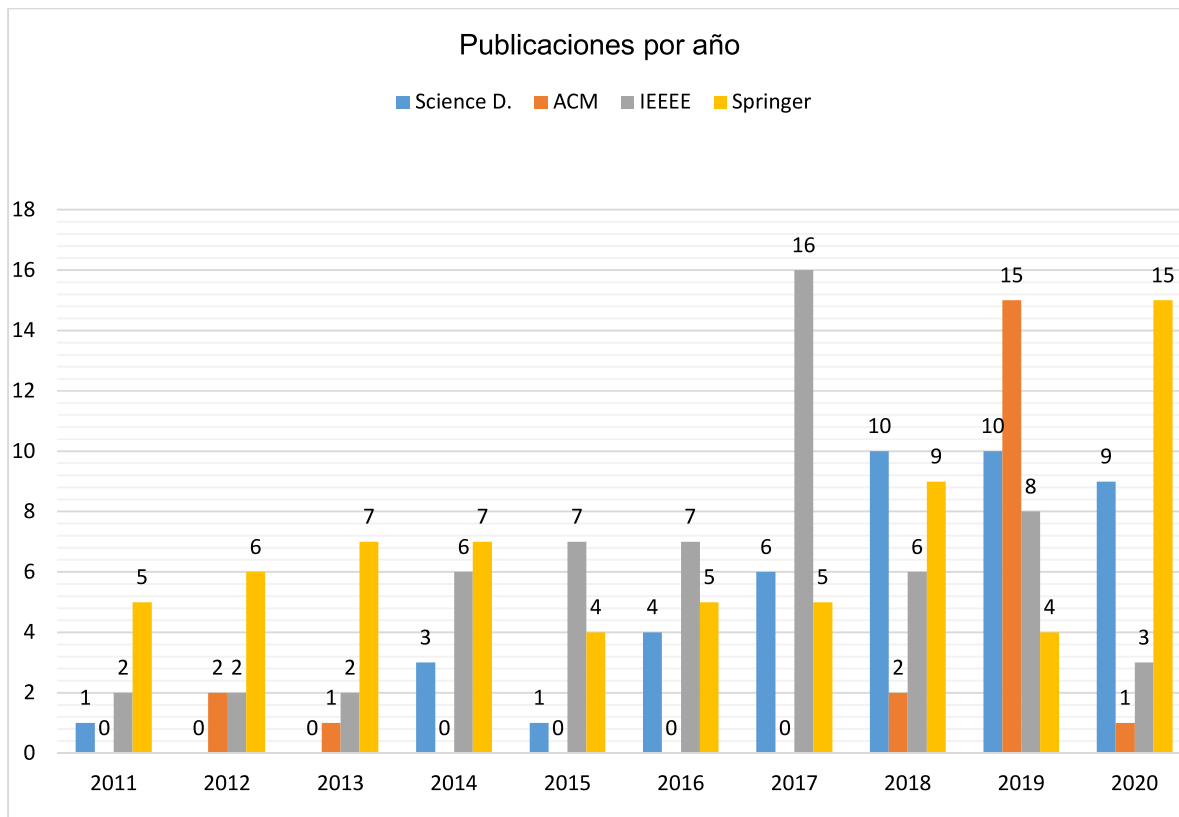


Figura 34. Publicaciones agrupadas por año de publicación.

2.3.3. Extracción y síntesis de datos.

Los trabajos seleccionados se concentran en el pre-procesamiento de imágenes, la extracción de características de una firma manuscrita y los métodos de clasificación en especial los basados en redes neuronales.

SQ1. ¿Cuáles son los métodos usados para el pre-procesamiento y extracción de características de imágenes de firmas offline?

En la fase de pre-procesamiento de firmas manuscritas, la imagen de la firma es preparada mejorando su calidad para que la extracción de características sea eficiente. Generalmente la firma se la transforma a una imagen en tonos de gris, algunos algoritmos convierten la imagen original en una imagen binaria [12], a estas imágenes se les aplica métodos de normalización para regular la rotación y la posición sin deformar las estructuras de la imagen, algunos trabajos aplican ecualización basada en el histograma de la imagen con el objeto de mejorar

los contrastes bajos de una imagen en tonos de gris [27], posteriormente se procede a la reducción de ruido presente en la imagen mediante la aplicación de filtros y operaciones de convolución sobre la imagen, los tipos de filtros a aplicar dependen del tipo de ruido que se encuentre en la misma [12] [14]. Se ha encontrado trabajos de investigación en los cuales además de binarizar la imagen es sometida a un proceso de esqueletización, método basado en la eliminación sucesiva de capas del contorno de la firma con el objetivo de obtener trazos de la firma cuyo grosor es de tamaño uno [12] [16], la idea es que esta imagen esqueletizada exprese la forma, orientación, inclinación, de los trazos de la firma.

Posterior al pre proceso de la imagen se aplica métodos de extracción de características de la firma para su clasificación. Se han encontrado una gran variedad de trabajos de investigación en relación con este tema. Existen métodos basados en la Transformada Discreta de Fourier, Transformada Discreta de Wavelet [28], en el caso de la transformada de Fourier esta es aplicada a segmentos de la imagen en donde se procede a calcular los coeficientes de la transformada de Fourier los cuales son tomados como las características de la imagen, luego estas características pueden ser extraídas de toda la imagen, En [29] hace referencia a la extracción de características globales y locales de la firma. Se mencionan como características globales a el peso de la firma; la relación entre el ancho y la altura de la imagen, que se calcula usando la razón matemática altura / ancho; la altura solamente; el ancho solamente; el área de la imagen calculada como el número de píxeles de color negro que se encuentran en el esqueleto de la imagen, la máxima proyección horizontal, la máxima proyección vertical; los picos de la proyección horizontal y vertical; centro de gravedad vertical y el centro de gravedad horizontal, número de puntos negros finales y número de puntos negros cruzados. Como características locales que representan a un segmento limitado la imagen de la firma se mencionan a la densidad de píxeles calculada como el número de píxeles negros o la proporción de éstos en cada segmento, distribución de píxeles o la distribución geométrica del pixel en una celda, distancia o características de los ángulos, entre otros. El trabajo de investigación [29] hace referencia de manera explícita al cálculo de la característica local "distancia o características de los ángulos", donde la esquina inferior izquierda de cada segmento es tomada como punto de origen (0,0) y la distancia es calculada con relación al origen de cada segmento de la imagen, la figura 35 muestra la idea de este cálculo.

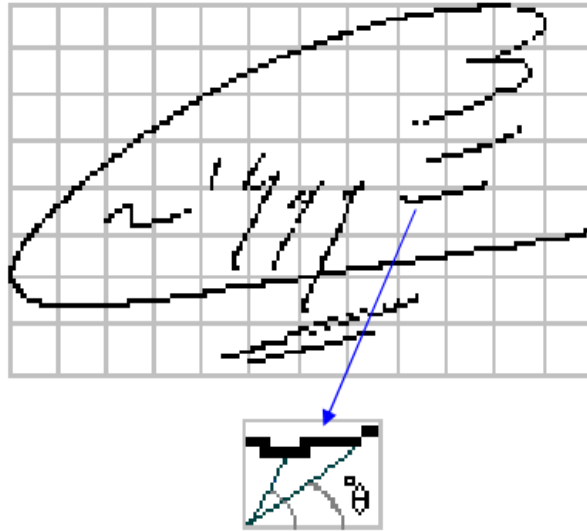


Figura 35. Extracción de la característica del ángulo [29].

Otro conjunto de características que se pueden tomar de una imagen son: La relación entre la longitud horizontal y diagonal, el radio de aspecto, la pendiente de la recta que une el centro de gravedad y el centro geométrico, la varianza vertical y horizontal de la firma, las señales envolventes obtenidas basándose en el centro de gravedad [6]. En resumen, existe una gran cantidad de trabajos que hacen referencia a la extracción de características de una imagen basándose especialmente en la información de la forma geométrica de la imagen.

SQ2. ¿Cuáles son los algoritmos de clasificación supervisada para imágenes?

Se ha encontrado varios enfoques de clasificación de imágenes de firmas manuscritas, Un enfoque interesante propone el cálculo basado en fractales. La teoría fractal fue propuesta por el matemático Mandelbrot en 1975, el principio básico sostiene que un objeto se considera fractal si tiene una propiedad de auto semejanza, es decir, si su forma puede ser repetida varias veces a diferentes escalas [30]. En este trabajo se propone el cálculo de la dimensión fractal de la imagen de una firma a pesar de que se considera que la firma no es un objeto fractal, se propone el cálculo del fractal de cada firma utilizando tres métodos, los valores encontrados son almacenados en un vector $V = (D1, D2, D3)$ que formaran parte del vector de características, los tres métodos propuestos para el cálculo del fractal son: Recuento de cajas, radio de masa y DLA, que consiste en elegir aleatoriamente n -píxeles que pertenecen a la estructura y contando cada píxel [30]. Otra propuesta incluye la combinación del uso de la Transformada Discreta de Radon (TDR) en combinación de una red neuronal probabilística

[31], TDR es usada para representar una firma desde diferentes ángulos, las imágenes bidimensionales son transformadas en líneas en donde, cada línea de la imagen dará un pico posicionados en los parámetros de la línea correspondiente, este cálculo genera una cantidad de valores demasiado grande, por lo que para la compresión de los datos se usa la técnica de Análisis de Componentes Principales(PCA), para la fase de clasificación se combina con una red neuronal probabilística de tres capas. En la figura 36 se muestra tres ejemplos de dos firmantes y su respectiva imagen relacionada con la transformada de Radón

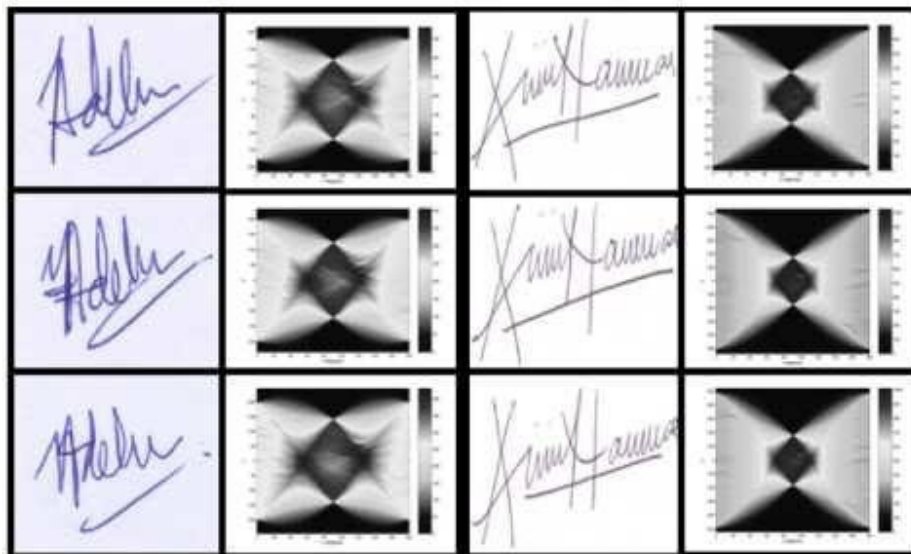


Figura 36. Transformada discreta de Radón de dos firmantes.

Un trabajo propone otro método de verificación de firmas basado en la aplicación de filtros de ondas complejas rotados (RCWF) combinada con la transformada de ondas complejas de árbol dual (DT-CWT), para extraer doce diferentes características de la imagen de la firma, el rendimiento de ese trabajo ha sido probado en una base de datos de 192 firmas, este trabajo es una variación de trabajos que usan DWT (transformada de wavelet), debido a que tiene dos desventajas, en primer lugar da una mala direccionalidad y presenta mucha sensibilidad al cambio [32]. Una visión diferente presenta [33], que se basa en el uso de Support Vector Machine (SVM), SVM intenta encontrar la separación óptima del hiperplano que separa los ejemplos positivos y negativos con un máximo margen, este trabajo combina SVM con descriptores de imagen basados en el histograma piramidal de gradientes orientados (PHOG). En el artículo de investigación [34], se estudia el problema de identificación de escritura a mano para identificar a la persona por sus hábitos de escritura, usando el modelo de redes generativas adversarias, esta arquitectura es un modelo de aprendizaje profundo que se basa

en aprender nuevas muestras de entrenamiento, consta de dos módulos, el módulo de generación de datos(G) y el módulo discriminante(D), en el proceso de entrenamiento el objetivo de G se generan muestras realistas para engañar a D, mientras que el objetivo de D es separar las muestras generadas por G de las muestras reales. En la figura 37, se muestra descripción didáctica de la arquitectura de la red generativa adversaria.

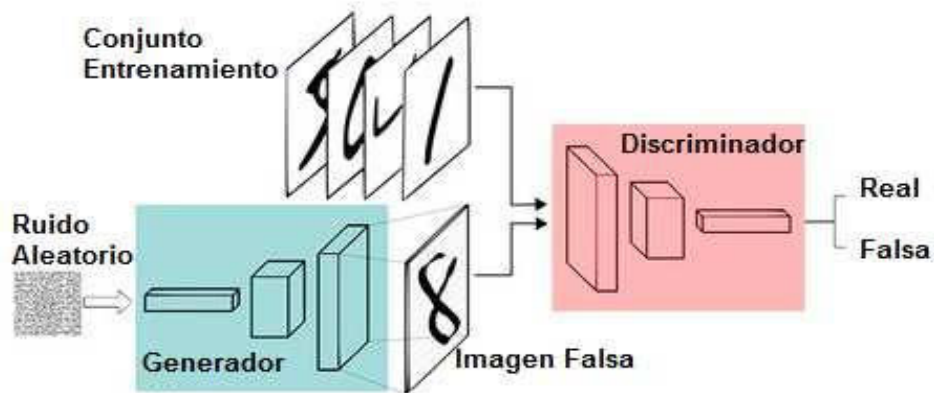


Figura 37. Red generativa adversaria.

SQ3. ¿Qué algoritmos basados en redes neuronales profundas existen para clasificación supervisada de imágenes?

Los trabajos más recientes hacen referencia al uso de redes neuronales profundas en especial a las redes neuronales convolucionales, que son las redes que más han tenido resultados satisfactorios en la clasificación de imágenes tanto de color como de imágenes en tonos de gris. En el trabajo de investigación [35] se propone un enfoque en dos etapas, la fase de aprendizaje de características de la firma independientes, seguida de una etapa de clasificación, este trabajo difiere en el trato de las imágenes, debido a que usa imágenes invertidas es decir el color blanco tiene un valor 0 y el negro tiene un valor de 1, para remover el ruido previo a la inversión de la imagen, usa el algoritmo OTSU's para obtener las intensidades optimas de los píxeles y después someter a la imagen a una normalización, se hicieron pruebas usando dos bases de datos GPDS-960 y Brazilian PUC-PR, el punto relevante de este trabajo es que se ha entrenado a la red para discriminar a firmas de usuarios no inscritos en el sistema de tal manera que no se confunda con los usuarios registrados en la red. Otro tipo de red convolucional es el uso de redes siamesas, son redes gemelas, este tipo de red es una clase de arquitectura que contiene dos subredes idénticas, tienen la misma

configuración, los mismos parámetros pero los pesos son compartidos, estas dos subredes están unidas por una función de pérdida en la parte superior que calcula una métrica de similitud que involucra a la distancia euclidiana entre la representación de características de cada lado de la red [36], el objetivo es determinar si una firma es genuina o es falsificada. En la figura 38, se presenta la arquitectura de la red siamesa.

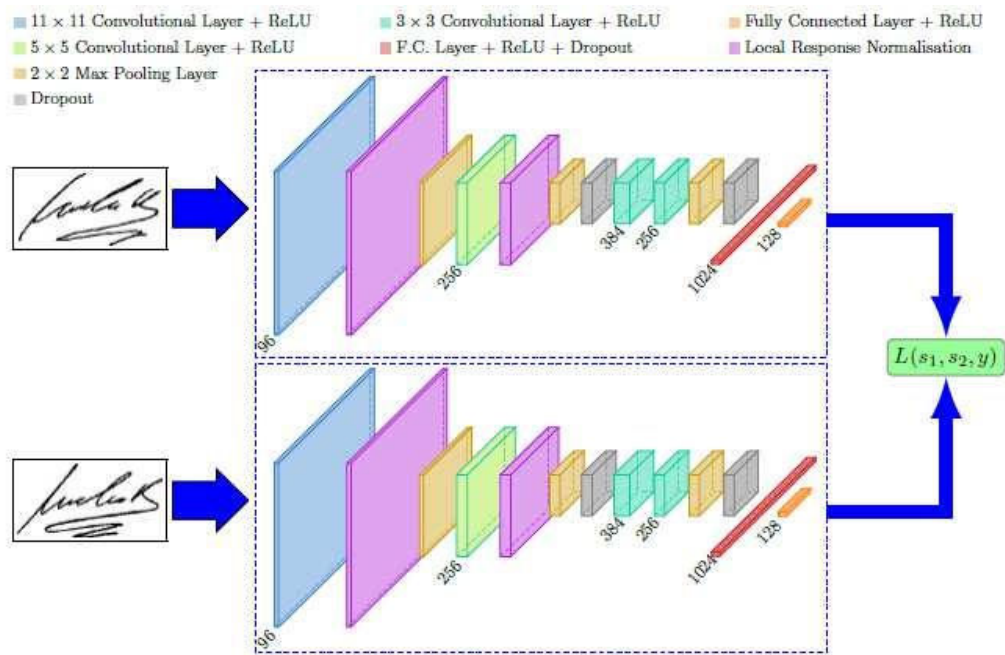


Figura 38. Red convolucional siamesa [36].

Para la evaluación del algoritmo se han considerado cuatro bases de datos, CEDAR, GPDS300, GPDS Synthetic Signature Database y BHSig 600, bases de datos con firmas genuinas y falsificadas, en el caso de CEDAR es una base de 55 firmantes en donde cada persona ha firmado 24 veces y cada firmante trata de hacer ocho imitaciones de tres personas.

Finalmente el trabajo de investigación [37], propone una red neuronal usada para verificación de firmas con una estructura de capas de convolucionales y de clasificación muy ligera, los datos muestran un accuracy entre el 90 y 98% con las bases de firmas usadas, es un algoritmo reciente de abril del 2020, en este trabajo se sostiene de que no es necesario una red convolucional compleja para lograr una buena clasificación de imágenes de firmas manuscritas.

3. MATERIALES Y MÉTODOS

En este trabajo se utilizó el método de investigación experimental, el cual se puede definir como “Un proceso planificado de investigar en donde una o más variables (variable independiente) es manipulada intencionalmente por el investigador para conocer los efectos que produce en al menos otra variable llamada dependiente” [38]. En la parte experimental se usó un enfoque cuantitativo mediante la recolección de datos y su análisis, para lograr este objetivo se usó la medición numérica y el análisis estadístico, con el fin de establecer patrones de comportamiento.

Las fases que se ejecutaron en esta investigación son las siguientes:

1. Revisión de literatura. Para el estudio del problema, se realizó una revisión de literatura en el tema de clasificación de firmas manuscritas utilizando métodos de aprendizaje supervisado, tema que se revisó en el capítulo anterior.
2. Adquisición de datos (Recolección de firmas). En esta fase se realizó la obtención de firmas manuscritas de personas residentes en el Ecuador, que luego servirá para la construcción de una base de datos de imágenes de firmas digitalizadas y se escogió adicionalmente otro dataset público llamado CEDAR.
3. Selección, adaptación e implementación en lenguaje Python de los algoritmos de clasificación supervisada a ser evaluados. En esta fase se realizó lo siguiente:
 - Selección de herramientas para la implementación de los algoritmos
 - Selección de los algoritmos de clasificación.
 - Implementación de los algoritmos en lenguaje Python.
4. Entrenamiento de los modelos escogidos y del modelo creado con los datasets especificados en el punto 2. En la figura 39, se muestra el esquema general de un clasificador de imágenes.
 - Se aplicó un pre-proceso de la imagen, con el objetivo de suprimir el ruido y normalizar la imagen, permitiéndonos trabajar con imágenes de tamaños similares.
 - Con la información obtenida en los pasos anteriores se entrenó los algoritmos de clasificación escogidos sobre los dos datasets de firmas manuscritas..

5. Evaluación y comparación de algoritmos. En esta fase se procedió a ajustar los algoritmos para aumentar la precisión de la clasificación y disminuir el error en la clasificación.
6. Análisis de datos. Se analizó los datos relacionados con el rendimiento de los algoritmos basado en el estudio de la tasa de error, de exactitud y matriz de confusión.
7. Discusión y conclusiones. Con los resultados obtenidos, se establecieron las conclusiones del estudio y se propuso trabajo futuro.

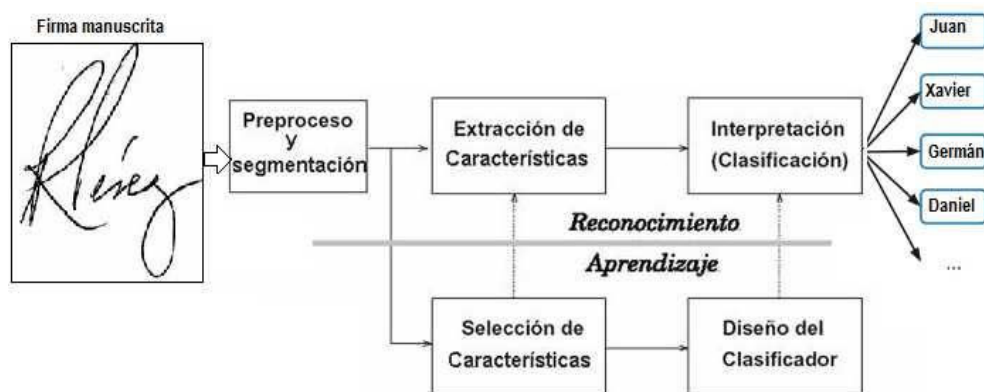


Figura 39. Esquema general de un clasificador de imágenes

3.1. ADQUISICIÓN DE DATOS.

En esta etapa se presentaron problemas en la adquisición de firmas manuscritas. Estos inconvenientes fueron dos claramente evidenciados: En primer lugar, se tiene, el uso de un tipo de papel blanco de distinto gramaje, para el registro de las firmas y en segundo lugar el formato del documento en donde se registraba la firma, este formato no definía un área claramente reservada para el registro de la firma.

Usar distintos tipos de papel provocó un problema, ya que no se pudo aplicar un método estándar para la reducción del ruido, debido a que el método usado para la reducción de ruido funcionaba en algunos casos y en otros no era eficientemente, fue necesario variar los parámetros del método de reducción de ruido y en otros casos se optó por cambiar el algoritmo de reducción del ruido. Este es un serio problema debido a que dependemos de un tipo de

papel, esto no es óptimo porque el método no se pudo generalizar para toda clase de papel blanco.

El segundo problema se presentó en la adquisición de las imágenes, debido a que el formato establecido inicialmente para la captura de firmas era ineficiente. Este formato era muy abierto, permitiendo que el usuario firme en un área no demarcada claramente, este particular provocó un problema al momento de procesar la firma, debido a que esta no se encontraba centrada y en algunos casos se encontraba cortada. (Esto se evidencia en la figura 41). Por esta razón se procedió a cambiar el formato de adquisición de firmas en donde se propone un área claramente demarcada y en donde el usuario debía registrar la firma. Las figuras 40, 41 muestran los problemas encontrados en la recolección de datos.

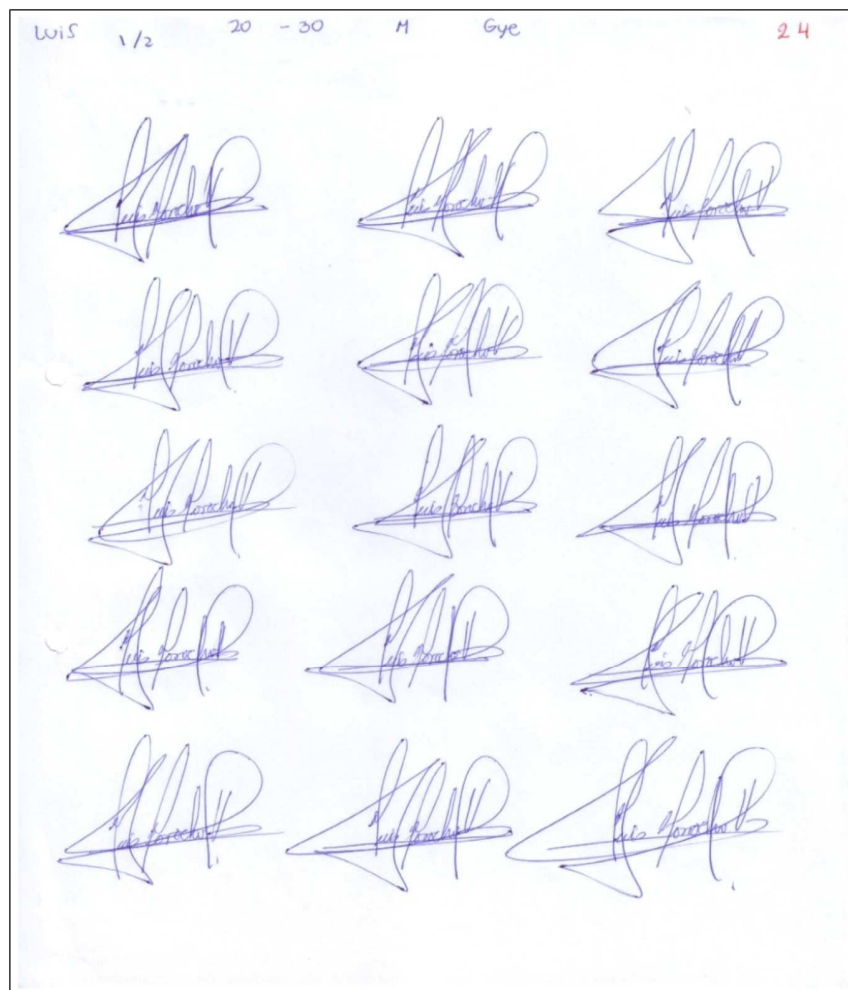


Figura 40. Formato de recolección de firmas ineficiente.

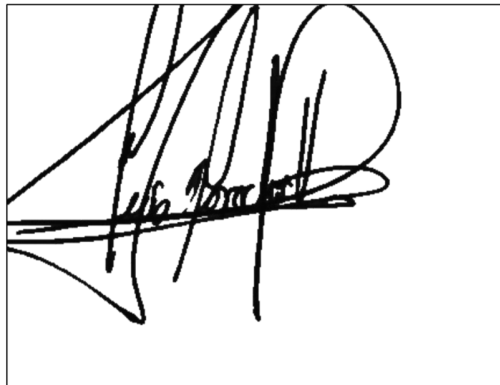


Figura 41. Problema de firma incompleta debido al uso de un formato libre.

99 Nombre: Vinicio C Edad: 37 Sexo: (M / F) Ciudad Residencia: Pachala

Figura 42. Formato final de recolección de firmas

La figura 42, muestra el formato final usado para la recolección de datos.

Con relación a la cantidad de datos, se procedió a obtener firmas de 175 personas. Estas están divididas en dos datasets. Para el primer dataset, se realizó una recolección de datos de firmas de manera personal, solicitando a 121 personas 45 firmas, estas fueron recolectadas en tres hojas en donde la persona firmó quince veces en cada hoja como se muestra en la figura 42, De estas 45 firmas, 31 firmas se usaron para entrenamiento, siete firmas para pruebas y siete firmas para la validación y verificación de los algoritmos, para referenciar a esta base de datos la nombraremos DB-GC. Adicionalmente se usó la base de datos CEDAR que es una base de datos de acceso público que consta de 55 firmantes en donde cada persona ha firmado 24 veces, de estas firmas, 20 han sido usadas para el entrenamiento y 4 firmas han sido utilizadas para la validación, la base CEDAR fue usada en el trabajo de investigación [36]. El objetivo de esto es hacer una comparación del rendimiento de los algoritmos seleccionados con dos bases de datos diferentes. En la figura 43 se muestra la estructura de la base de datos DB-GC, agrupada por rango de edad.

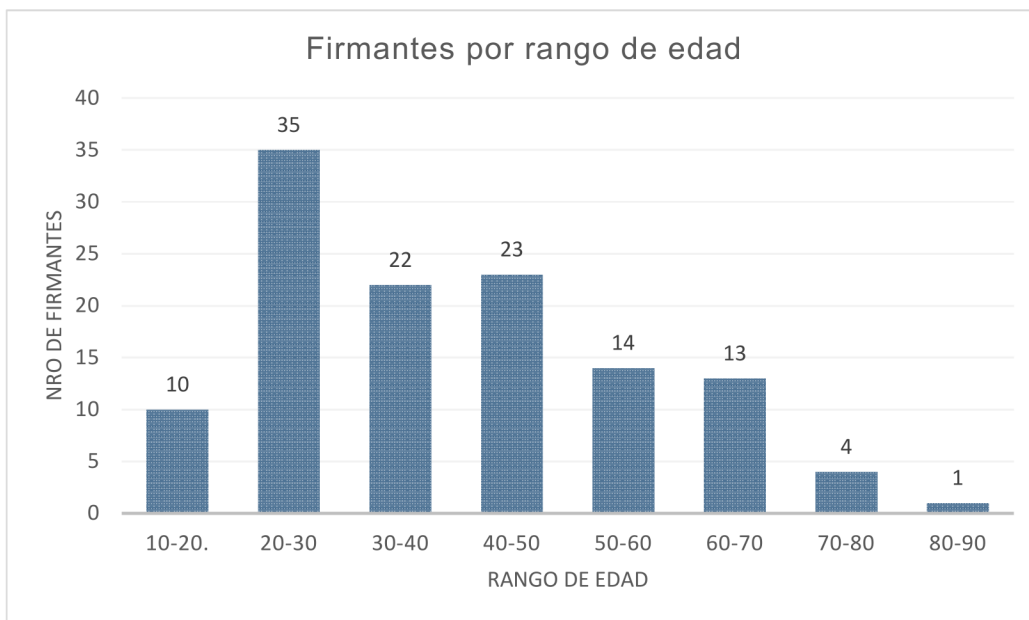


Figura 43. Firmantes por rango de edad de la base de firmas DB-GC.

3.2. SELECCIÓN, ADAPTACIÓN E IMPLEMENTACIÓN DE ALGORITMOS.

En esta sección se explican las decisiones que se tomaron con relación a los algoritmos a usar, y el software que se empleó.

3.2.1. Selección de herramientas de desarrollo.

Las herramientas seleccionadas para este trabajo se describen a continuación.

- Programación, debido a la gran aceptación en aplicaciones de inteligencia artificial, por ser un lenguaje de alto nivel y al conocimiento y experiencia personal que se tiene de esta herramienta se escogió el lenguaje PYTHON, en su versión 3.7.
- TensorFlow, es una librería de código abierto que se ha usado en aprendizaje automatizado, esta librería es desarrollada por Google y se puede integrar fácilmente con los proyectos realizados en Python.
- Keras, un API de alto nivel escrito en lenguaje PYTHON y creado sobre TensorFlow, para la construcción de redes neuronales con mayor facilidad, es muy usado en aplicaciones de Machine Learning y Deep Learning.
- SciKit Learn, un API open source, escrita en lenguaje PYTHON.
- Open CV (Open Computer Vision), versión 4.1, una librería de código abierto escrita en múltiples lenguajes de programación como C, C++, Java y Python entre otros. Esta librería está orientada a aplicaciones de visión artificial y proporciona un entorno fácil de usar y muy eficiente en el tratamiento y manejo de imágenes.
- Dash, escrito en Python es un framework que permite construir aplicaciones analíticas orientas a la web, con interfaces intuitivas.

3.2.2. Selección de algoritmos de clasificación.

Los algoritmos de clasificación seleccionados están basados en redes neuronales convolucionales, el primer algoritmo es de autoría propia, el segundo algoritmo está basado en una red convolucional que inicialmente se ha aplicado a reconocimiento de dígitos y la tercera arquitectura está estrictamente orientado a reconocer firmas manuscritas, el último algoritmo llamado red convolucional poco profunda, fue publicado en abril del 2020 [37].

El hecho de escoger algoritmos basados solamente en redes neuronales convolucionales, se debe a que al analizar los algoritmos de este tipo se encontró que eran los de mejor rendimiento para este tipo de aplicaciones, además existen librerías de alto nivel que están ya maduras en ésta área. Adicionalmente desde el punto de vista computacional, enfocándonos solamente en el uso de estos algoritmos, podemos encontrar que son simples y livianos de usar, que es lo que se busca como uno de los objetivos de este trabajo. Un tercer aspecto es el deseo del autor de profundizar en el estudio de este tipo redes. Los algoritmos escogidos se describen a continuación.

Red Neuronal CN-GC. Esta red fue nombrada CN-GC, debido a que GC son las iniciales del nombre del autor de este trabajo, por ser este modelo de creación propia y las iniciales CN provienen de convolucional network.

Esta red neuronal convolucional, es una red construida en base a la experiencia de prueba y error, de autoría propia y no ha sido presentada previamente en ningún trabajo de investigación, nació en las aulas de la universidad. Inicialmente dio resultados de exactitud muy pobres, por lo que progresivamente se ha modificado, probando cada vez con una mayor cantidad de datos de firmantes, poco a poco se cambió el modelo consiguiendo mejores resultados, hasta que se ha llegado a superar el valor de 97% de exactitud en la fase de entrenamiento. Este proceso de creación y prueba tomó aproximadamente diez meses. La idea principal es comparar este modelo con otras redes neuronales convolucionales y de manera progresiva ir mejorando la arquitectura de la red mientras se va adquiriendo experiencia. El objetivo es tener un modelo que se pueda usar en trabajos futuros sin entrar en conflictos por el tema de los derechos de autor.

La arquitectura de la red se detalla en la tabla 5:

Capa	Tamaño kernel	Filtros y avance	Función activación	Descripción
Conv2D 340x440	5x5	32, 1	RELU	Capa convolucional con imágenes de 340 x 440, el tamaño de la matriz de convolución es de 5x5, se aplican 32 filtros con avances de 1 pixel cada uno, la función de activación es RELU.
Conv2D 340x440	5x5	16, 1	RELU	Capa convolucional con imágenes de 340 x 440, el tamaño de la matriz de convolución es de 5x5, se aplican 16 filtros con avances de 1 pixel cada uno, la función de activación es RELU.
MaxPooling	2x2	0	MAX	Capa de pooling con una entrada de imágenes de 340 x 440, la dimensión del filtro es de 2x2, la función de pooling que se aplica es el máximo.
Conv2D 170x220	3x3	12, 1	RELU	Capa convolucional con imágenes de 170 x 220, matriz de convolución de 3x3, 12 filtros y avances de 1 pixel en cada filtro, la función de activación es RELU.
MaxPooling	2x2	0	MAX	Capa de pooling con una entrada de imágenes de 170 x 220, la dimensión del filtro es de 2x2, la función de pooling que se aplica es el máximo.
Conv2D 85x110	3x3	8, 1	RELU	Capa convolucional imágenes de 85 x 110, la dimensión de la matriz de convolución es de 3x3, se aplican 8 filtros con avances de 1

				pixel cada uno, la función de activación es RELU.
MaxPooling	2x2	0	MAX	Capa de pooling con una entrada de imágenes de 85 x 110, la dimensión del filtro es de 2x2, la función de pooling que se aplica es el máximo.
Conv2D 42x55	3x3	8, 1	RELU	Capa convolucional con una entrada de imágenes de 42 x 55, la dimensión de la matriz de convolución es de 3x3, se aplican 8 filtros con avances de 1 pixel cada uno, la función de activación es RELU
MaxPooling	3x3	0	MAX	Capa de pooling con imágenes de 42 x 55, la dimensión del filtro es de 3x3, la función de pooling que se aplica es el máximo
Faltten				Capa de aplanamiento en donde se inicia la capa de clasificación
Dense 1x256	0.0			Una capa oculta de 256 unidades o neuronas con un %0 de drop out
Dense 1x256	0.0			Una capa oculta de 256 unidades con un %0 de drop out.
Dense 1xn	0.0			Una capa de n unidades, donde n es el número de firmantes de la base CEDAR o DB-GC

Tabla 5. Descripción red convolucional CN-GC.

Red Neuronal Reconocimiento de dígitos. La arquitectura de esta red ha sido publicada en la revista The Journal of Computing Sciences in Colleges [39]. El uso de esta arquitectura de red convolucional orientada al reconocimiento de dígitos manuscritos para compararla con arquitecturas orientadas al reconocimiento de firmas, se basa en el concepto de que un dígito también tiene rasgos caligráficos al igual que una firma manuscrita, un poco más sencillos probablemente, se puede plantear que las firmas podrían analizarse como dígitos que no van de 0 al 9, sino de 0 a 120 (tamaño de la base de datos DB-GC), se desea verificar que tan eficiente es esta arquitectura de red convolucional al aplicarla en imágenes de firmas manuscritas.

Capa	Tamaño kernel	Filtros y avance	Función activación	Descripción
Conv2D 340x440	3x3	32, 1	RELU	Capa convolucional con una entrada de imágenes de 340 x 440, el tamaño de la matriz de convolución es de 3x3, se aplican 32 filtros con avances de 1 pixel cada uno, la función de activación es RELU.
Conv2D 28x28	3x3	16, 1	RELU	Capa convolucional con una entrada de imágenes de 28 x 28, la dimensión de la matriz de convolución es de 3x3, 16 filtros y avances de 1 pixel en cada filtro, la función de activación es RELU.
MaxPooling	3x3	0	MAX	Capa de pooling con una entrada de imágenes de 28 x 28, la dimensión de la matriz de convolución es de 3x3, la función de maxpooling.
Conv2D 14x14	2x2	12, 1	RELU	Capa convolucional con de imágenes de 14 x 14, la dimensión de la matriz de convolución es de 2x2, se aplican 12 filtros con

				avances de 1 pixel cada uno, la función de activación es RELU.
MaxPooling	3x3	0	MAX	Capa de pooling con una entrada de imágenes de 14 x 14, la dimensión de la matriz de convolución es de 3x3, la función de maxpooling.
Faltten				Capa de aplanamiento en donde se inicia la capa de clasificación.
Dense 1x120		0.25		Una capa oculta de 120 unidades, con un dropout de 25%.
Dense 1x84		0.25		Una capa oculta de 84 unidades, con un dropout de 25%.
Dense 1xn		0.25		Una capa de salida de n unidades, donde n es la cantidad de firmantes (clases) en la base de datos DB-GC y CEDAR.

Tabla 6. Estructura de la red convolucional reconocimiento dígitos [39].

Red neuronal poco profunda. El trabajo de investigación [37], propone un modelo de red neuronal convolucional que la han denominado red convolucional poco profunda, propone la hipótesis de que las redes neuronales convolucionales no necesariamente tienen que ser muy complejas, en cuanto al número de capas convolucionales ni tampoco en su capa clasificadora, para realizar un buen trabajo de clasificación, de este concepto nace su nombre red neuronal poco profunda. Los datos presentados en el trabajo de investigación [37] indican una exactitud (accuracy) entre el 90% al 98% de exactitud, este modelo fue entrenado en 9 bases de firmas. La estructura de la red se muestra en la tabla 7.

Capa	Tamaño kernel	Filtros y avance	Func. Activ.	Descripción
Conv2D 128x64	7x7	40, 1	RELU	Capa convolucional con imágenes de 128 x 64, el tamaño de la matriz de convolución es de 7x7, se aplican 40 filtros con avances de un pixel cada uno, la función de activación es RELU.
MaxPooling	2x2	0	MAX	Capa de pooling con una entrada de imágenes de 128 x 64, la dimensión de la matriz de convolución es de 2x2, la función de pooling que se aplica es el máximo.
Conv2D 61x29	5x5	30, 1	RELU	Capa convolucional con imágenes de 61 x 29, la dimensión de la matriz de convolución es de 5x5, se aplican 30 filtros con avances de un pixel cada uno, la función de activación es RELU.
MaxPooling	3x3	0	MAX	Capa de pooling con una entrada de imágenes de 61 x 29, la dimensión de la matriz de convolución es de 3x3, la función de pooling que se aplica es el máximo.
Conv2D 19x8	3x3	20, 1	RELU	Capa convolucional con imágenes de 19 x 8, la dimensión de la matriz de convolución es de 3x3, se aplican 30 filtros con avances de un pixel cada uno, la función de activación es RELU.
MaxPooling	3x3	0	MAX	Capa de pooling con una entrada de imágenes de 19 x 8, la dimensión de la matriz de convolución es de 3x3, la función

		de pooling que se aplica es el máximo.
Faltten		Capa de aplanamiento en donde se inicia la capa de clasificación.
Dense 2xn+d	0.0	Una capa oculta de $2 \times n + d$, en donde n es el número de clases y d es el número de dimensiones en este caso es 2.
Dense 1 x n	0.0	Una capa oculta de n unidades, donde n es el número de clases.

Tabla 7. Red neuronal convolucional poco profunda.

3.2.3. Pre proceso de la imagen de una firma.

El pre-procesamiento de la imagen de una firma manuscrita, tiene como objetivo eliminar ciertas características de la imagen que pueden influir de manera negativa en la interpretación posterior de esta imagen, entre estas características está el ruido inherente a los dispositivos de captura, al tipo de papel empleado, la iluminación del medioambiente, entre otros [12] .

En esta etapa se presentaron problemas en la adquisición de las firmas, estos inconvenientes fueron dos claramente evidenciados: En primer lugar, tenemos el uso de un tipo de papel blanco de distinto gramaje, para el registro de las firmas y en segundo lugar el formato del documento en donde se registraba la firma, este formato no definía un área claramente reservada para el registro de la misma.

Los métodos de reducción de ruido usados son el método gaussiano, de la mediana y ecualización de imagen. Se hizo pruebas con métodos como la aplicación de la Transformada Inversa de Fourier y la aplicación de operaciones morfológicas, pero estos no resultaron efectivos. En el caso de la transformada inversa de Fourier, este método es más eficiente cuando el ruido a ser eliminado sigue un patrón basado en la frecuencia, que no es el presente caso.

También se probó como parte de la fase de pre-proceso, el método de esqueletización de firmas como el último paso del pre-proceso, al proceder con el entrenamiento de los tres modelos se obtuvieron datos muy pobres en todas las métricas de evaluación.

Épocas	Exactitud		Error medio cuadrado(EMC)		Pérdida(loss)	
	Valid	Entren.	Valid	Entrenam.	Loss-Val.	Loss-Entr.
1	0.018319	0.010835	0.017851	0.017852	4.008109	4.008449
2	0.017991	0.011349	0.017851	0.017852	4.007598	4.007984
3	0.017991	0.011980	0.017851	0.017852	4.006063	4.008008
4	0.017664	0.011475	0.017851	0.017852	4.007339	4.007984
5	0.018319	0.011465	0.017851	0.017852	4.008073	4.007975
6	0.017337	0.011475	0.017851	0.017852	4.007220	4.007979
7	0.017991	0.014111	0.017851	0.017852	4.006568	4.007984
8	0.017991	0.018143	0.017851	0.017851	4.008099	4.007432
9	0.018068	0.018301	0.017851	0.017851	4.007999	4.007390

Tabla 8. Datos de entrenamiento, algoritmo CN-GC, base CEDAR esqueletizada.

La tabla 8, muestra los valores del entrenamiento del algoritmo CN-GC usando la base CEDAR con el método de esqueletización, como se puede ver, la exactitud no superó el 1.8% y el algoritmo de entrenamiento se detuvo en la novena época, debido a que el entrenamiento ya no iba a mejorar.

Épocas	Exactitud		Error medio cuadrático(EMC)		Pérdida (loss)	
	Valid	Entrena.	Valid	Entrenam.	Loss-Val.	Loss-Entr.
1	0.018319	0.014660	0.017851	0.017852	4.007352	4.008904
2	0.017991	0.011179	0.017851	0.017852	4.006764	4.008017
3	0.018646	0.010099	0.017851	0.017852	4.006732	4.008005
4	0.017337	0.013054	0.017851	0.017852	4.006815	4.007971
5	0.017664	0.011034	0.017851	0.017852	4.007553	4.007980
6	0.018646	0.010246	0.017851	0.017852	4.006039	4.007989
7	0.017991	0.014029	0.017851	0.017852	4.006797	4.007974
8	0.017991	0.011192	0.017851	0.017852	4.008017	4.007993
9	0.018397	0.014643	0.017851	0.017852	4.008440	4.007996
10	0.018319	0.010561	0.017851	0.017852	4.007089	4.007998
11	0.018319	0.017970	0.017851	0.017851	4.007260	4.007415
12	0.017664	0.018265	0.017851	0.017851	4.006652	4.007397

Tabla 9. Datos de entrenamiento, tercer algoritmo con la base CEDAR esqueletizada.

En la tabla 9 se puede observar que la exactitud apenas superó el 1.7%, el EMC no baja y la función de pérdida tampoco mejora. Estos datos corresponden al tercer algoritmo con la base de datos CEDAR esqueletizada. El método de EarlyStopping en todos los intentos de entrenamiento detuvo la ejecución, sin superar estos datos muy pobres. Comportamientos similares se obtuvo al proceder con el entrenamiento usando el segundo algoritmo. Por esta razón se abandonó el uso del método de esqueletización como parte del pre-proceso de la imagen de una firma manuscrita.

Como se explicó en el segundo párrafo, fue necesario usar una combinación del método de la aplicación de un filtro de la media y el método basado en la ecualización del histograma de la imagen. El método de eliminación de ruido usado funciona en todo tipo de papel blanco, el problema que se encontró es que en papeles de gramaje muy delgado se tuvo que hacer ajustes en la matriz de la mediana para posteriormente usar el método de ecualización del histograma de la imagen para mejorar la eliminación de ruido.

Las matrices usadas son variaciones de la mediana. En el caso de las firmas que estaban en papel de poco gramaje se usó la siguiente variación.

$$\frac{1.00}{7.40} * \begin{array}{|c|c|c|c|c|} \hline 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ \hline 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ \hline 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ \hline 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ \hline 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ \hline \end{array}$$

Tabla 10. Matriz de mediana modificada, papel de gramaje delgado.

Para el caso de papel de gramaje normal la matriz de mediana es la normal y es la siguiente.

$$\frac{1.00}{9.00} * \begin{array}{|c|c|c|c|c|} \hline 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ \hline 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ \hline 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ \hline 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ \hline 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ \hline \end{array}$$

Tabla 11. Matriz de mediana normal para eliminación de ruido.

También se debe mencionar que la base de datos DB-GC, tenía 130 firmantes, de los cuales se tuvieron que eliminar a nueve firmantes porque se detectó firmas forzadas, firmas que no se pudieron recortar ya que el firmante firmó de manera desordenada. Otro punto importante es que en la base de datos DB-GC se incorporó firmas de dos personas con la enfermedad de Parkinson, enfermedad que de manera general produce temblores en las extremidades, resultando con imágenes de firmas irregulares, estas firmas fueron perfectamente clasificadas por los tres algoritmos. Por respeto a dichas personas no se menciona la clase en las que fueron colocadas.

3.3. ENTRENAMIENTO Y VALIDACIÓN DE ALGORITMOS.

Para el entrenamiento de estos modelos se creó una estructura de datos para las imágenes de firmas. En la figura 44, se describe la estructura de datos construida para los datasets de firmas DB-GC y CEDAR. Se usó una estructura de datos similar, variando solamente en la cantidad de firmantes de 121 y 55 respectivamente, con una cantidad de firmas por cada firmante de 45 y 24 respectivamente. Esta estructura de datos se basa en una estructura de directorios o carpetas en donde se almacenan archivos en formato png, en donde cada archivo contiene la imagen de una firma, la estructura de directorios es similar a la figura 44.

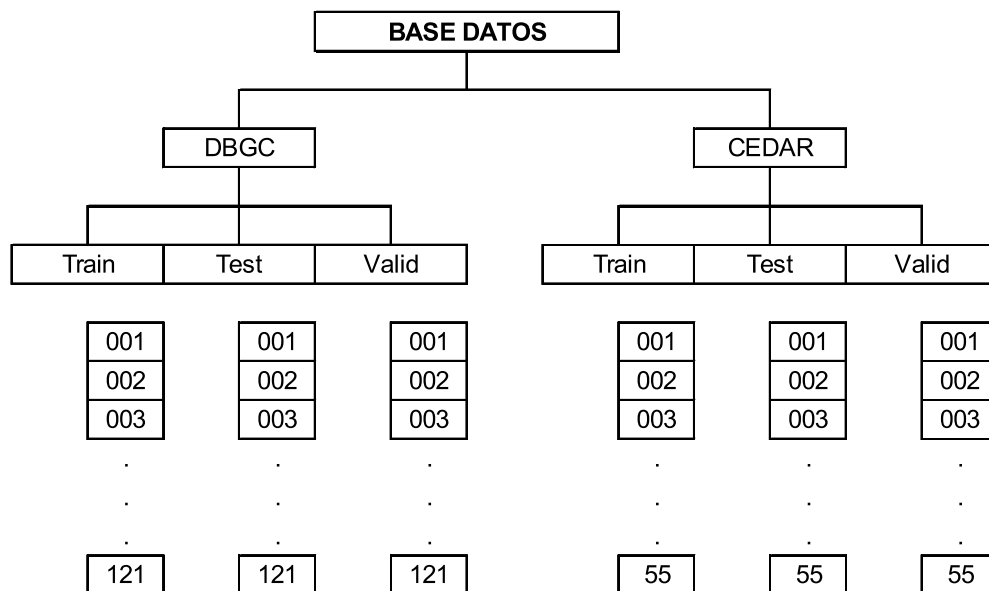


Figura 44. Estructura de directorios para las imágenes de firmas.

En la figura 44, se muestra dos bases de firmas DB-GC y CEDAR, para el dataset DB-GC, tenemos los ítems enumerados como 001, 002, 003,..., 121, estos son directorios en donde

se encuentra treinta y un firmas para el entrenamiento (train), siete firmas para la prueba (test) y siete firmas para la validación (valid), en el caso del dataset DB-GC, cada número identifica un firmante. De manera similar se puede observar para el dataset CEDAR en donde tenemos los ítems enumerados como 001, 002, 003,..., 55, estos son directorios en donde se encuentra diecisiete firmas para el entrenamiento (train), cuatro firmas para la prueba (test) y cuatro firmas para la validación (valid).

Con el propósito de incrementar datos en la fase de entrenamiento sin necesidad de requerir más firmas por cada usuario, se usó la técnica de aumento de datos (data augmentation), para lograr este objetivo, se aplican algoritmos mediante el uso de operaciones morfológicas sobre la imagen de la firma [40, 41]. Este tipo de operaciones son provistas de manera implícita por la herramienta Keras, que se usa para la programación de la red convolucional.

La información producto del entrenamiento de los modelos y gráficos relacionados con las métricas de exactitud, error medio cuadrático y pérdida, obtenidos en la fase de entrenamiento y los datos de las métricas de fase de pruebas se muestran y describen en la sección 4, en donde se presentan los resultados y comparación de estos modelos.

4. RESULTADOS Y DISCUSION.

En esta sección se presentan los resultados producto del entrenamiento de los tres algoritmos sobre los dos datasets CEDAR y DB-GC, los datos obtenidos en la fase de entrenamiento y validación, luego se procede a realizar comparaciones y resúmenes para su discusión.

El objetivo de este estudio de realizar una comparación de estos algoritmos no es determinar un ganador ni perdedor, el propósito es estudiar el comportamiento de cada algoritmo frente a otro, adquirir un conocimiento más profundo de este tipo red neuronal, implementar y evidenciar dicho comportamiento.

4.1. RESULTADOS DEL ENTRENAMIENTO.

En esta sección se presenta datos de los resultados producto del entrenamiento de los dos algoritmos escogidos y del algoritmo construido CN-GC, sobre los dos datasets de firmas manuscritas CEDAR y DB-GC.

4.1.1. Resultados del entrenamiento del algoritmo CN-GC, base CEDAR.

Los resultados del entrenamiento del primer algoritmo con la base de firmas CEDAR se presentan en la tabla 12, como se puede apreciar en los datos, la exactitud en el entrenamiento llegó al 99.73 %, siendo necesarias 15 épocas con 500 pasos por cada época, tomó un tiempo aproximado de 4 horas. Gráficamente la figura 45 muestra la exactitud del entrenamiento, la figura 46 el error medio cuadrático de la fase de entrenamiento y la figura 47 muestra los valores de la función de pérdida, estos datos fueron tomados de la tabla 12.

La herramienta Keras permite implementar una detención anticipada del entrenamiento, mediante el uso del método EarlyStopping, en donde se puede especificar una medida de rendimiento a ser monitoreada, mediante la cual se puede detener el entrenamiento cuando este ya no cumpla con el límite de control de esta medida, en el entrenamiento de todos los modelos se usó este método para evitar un overfitting de la red convolucional.

Épocas	Exactitud		Error medio cuadrático (EMC)		Pérdida (loss)	
	Validación	Entrenamiento	Validación	Entrenam.	Loss-Val.	Loss-Entre.
1	0.82630	0.36393	0.00452	0.01361	0.50144	2.38850
2	0.93360	0.81542	0.00176	0.00488	0.44336	0.63560
3	0.94864	0.89883	0.00136	0.00275	0.30121	0.34113

4	0.94472	0.92831	0.00154	0.00189	0.07580	0.22789
5	0.97089	0.94431	0.00084	0.00156	0.00532	0.18972
6	0.96238	0.95309	0.00105	0.00126	0.08713	0.14614
7	0.98070	0.95700	0.00050	0.00120	0.06938	0.14215
8	0.98855	0.96031	0.00036	0.00111	0.09956	0.13056
9	0.98160	0.97014	0.00052	0.00078	0.00047	0.08902
10	0.98463	0.96923	0.00039	0.00083	0.01833	0.09471
11	0.97579	0.97011	0.00069	0.00081	0.01882	0.09074
12	0.92509	0.97064	0.00218	0.00081	0.00754	0.09319
13	0.98626	0.97364	0.00040	0.00073	0.02567	0.08676
14	0.99640	0.99118	0.00010	0.00025	0.00155	0.02930
15	0.99738	0.99243	0.00010	0.00021	0.00125	0.02633

Tabla 12. Datos del entrenamiento, algoritmo CN-GC, base de firmas CEDAR.

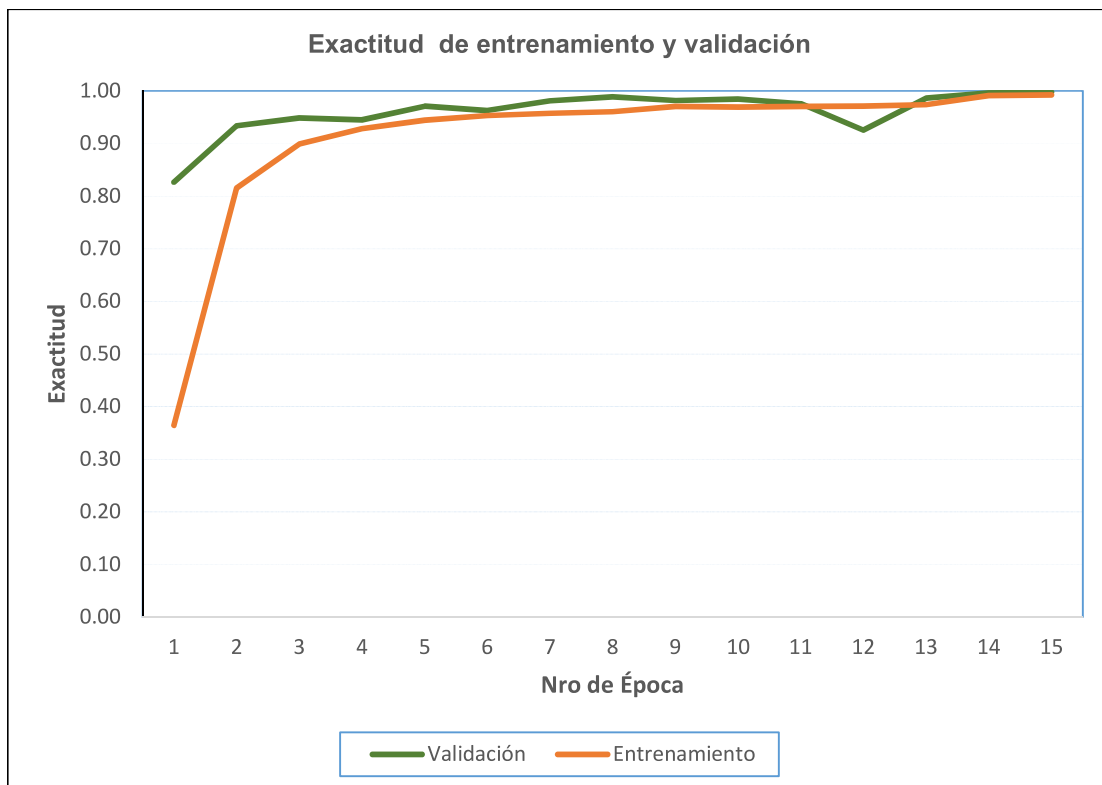


Figura 45. Exactitud de entrenamiento y validación, algoritmo CN-GC, CEDAR.

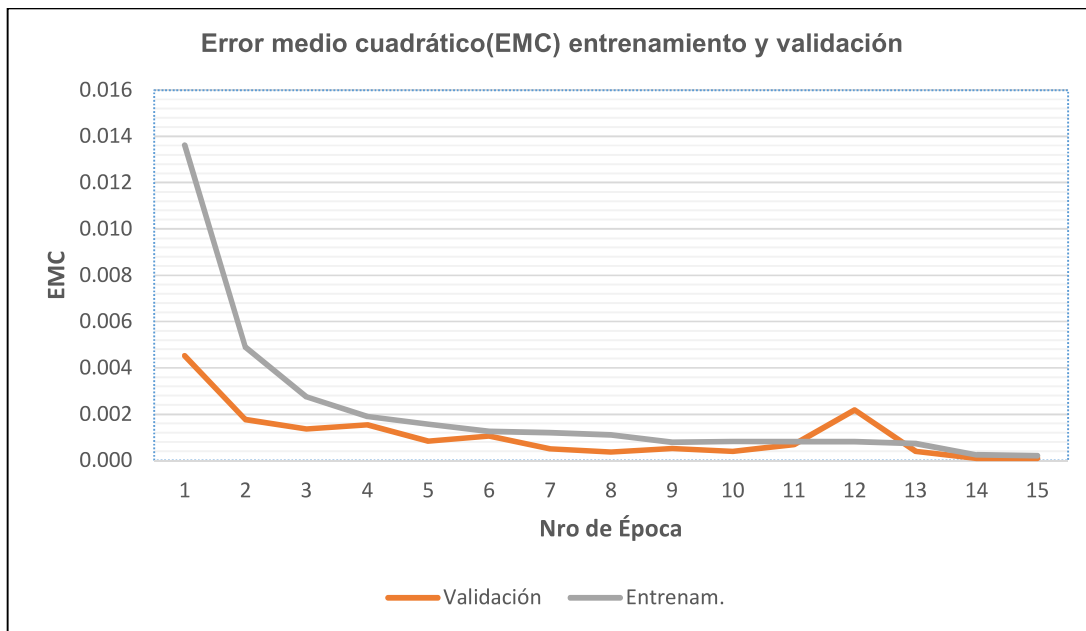


Figura 46. EMC, entrenamiento y validación, algoritmo CN-GC, dataset CEDAR.

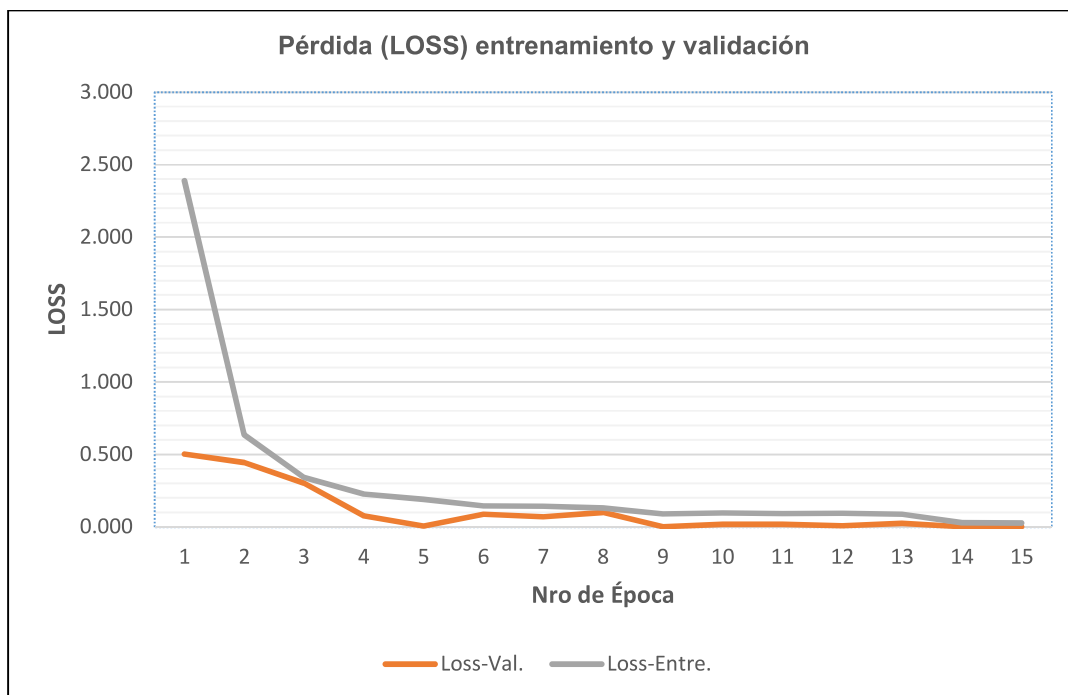


Figura 47, LOSS, entrenamiento y validación, algoritmo CN-GC, dataset CEDAR.

Posterior a la fase de entrenamiento se procedió a ejecutar el algoritmo CN-GC con datos de prueba, los resultados se presentan en la tabla 13. Se puede observar en los datos, que la

exactitud fue 98.18%, en contraste en el entrenamiento se alcanzó un 99.73% con la base de datos de firmas CEDAR.

MODELO CN-GC, BASE CEDAR					
CLASES	PRES	SEN	ESP	TFP	F1
1	1	1	1	0	1
2	1	1	1	0	1
3	1	1	1	0	1
4	1	1	1	0	1
5	1	1	1	0	1
6	1	1	1	0	1
7	1	1	1	0	1
8	1	1	1	0	1
9	1	0.8	1	0	0.8889
10	1	1	1	0	1
11	1	1	1	0	1
12	1	0.8	1	0	0.8889
13	1	1	1	0	1
14	1	1	1	0	1
15	1	1	1	0	1
16	0.75	1	0.9954	0.0046	0.8571
17	1	1	1	0	1
18	1	1	1	0	1
19	1	1	1	0	1
20	1	1	1	0	1
21	1	1	1	0	1
22	1	1	1	0	1
23	1	1	1	0	1
24	1	1	1	0	1
25	1	1	1	0	1
26	1	1	1	0	1
27	1	1	1	0	1
28	1	1	1	0	1
29	1	1	1	0	1
30	1	1	1	0	1
31	1	1	1	0	1
32	1	1	1	0	1
33	1	1	1	0	1
34	1	0.8	1	0	0.8889
35	1	1	1	0	1
36	1	1	1	0	1
37	1	1	1	0	1
38	1	1	1	0	1
39	1	1	1	0	1
40	1	1	1	0	1
41	0.75	1	0.9954	0.0046	0.8571
42	1	1	1	0	1
43	1	1	1	0	1
44	1	1	1	0	1
45	1	1	1	0	1
46	1	1	1	0	1
47	1	1	1	0	1
48	1	1	1	0	1
49	1	1	1	0	1
50	0.75	1	0.9954	0.0046	0.8571

51	1	1	1	0	1
52	0.75	1	0.9954	0.0046	0.8571
53	1	0.8	1	0	0.8889
54	1	1	1	0	1
55	1	1	1	0	1
Exact	0.9818	0	0	0	0
Prom	0.98181818	0.9854545	0.999665	0.000335	0.981527
DesvSt	0.06492208	0.0519377	0.001195	0.001195	0.045184

Tabla 13. Resultados fase de prueba, algoritmo CN-GC, base de datos CEDAR.

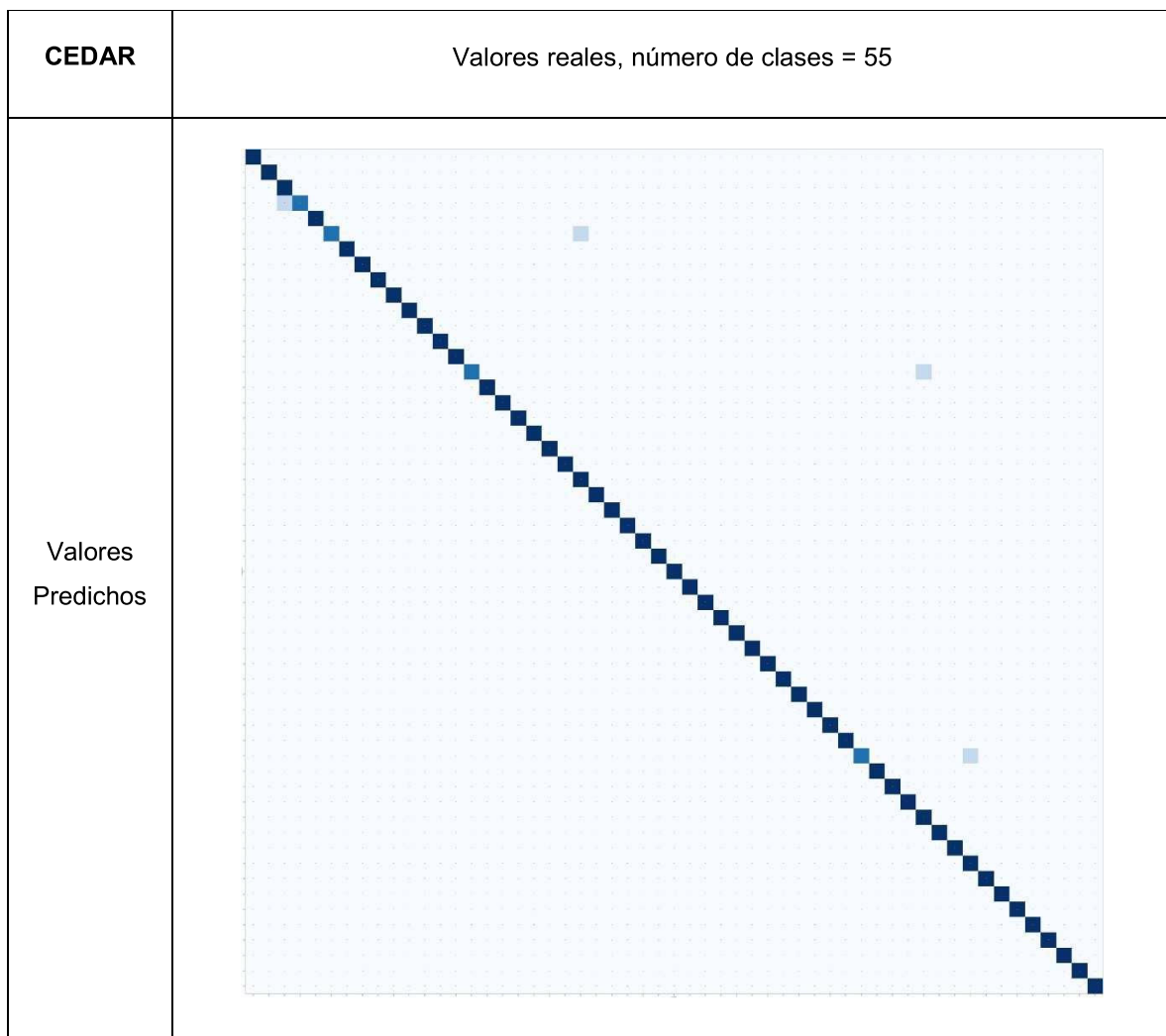


Figura 48. Matriz de confusión, primer algoritmo, para la base de datos CEDAR.

En la figura 48, se muestra la matriz de confusión para la base de datos CEDAR que tiene 55 clases. Cada celda pintada de azul oscuro en la diagonal indica una validación al 100%, las

celdas que están fuera de la diagonal, pintadas de un color azul claro, indican que en esa clase se produjo una validación inferior al 99%. Se puede apreciar que existen problemas solo en 4 clases, en las demás clases de esta base de datos, la predicción fue correcta.

4.1.2. Datos de entrenamiento del algoritmo CN-GC, dataset DB-GC.

Para la base de firmas DB-GC, en el proceso de entrenamiento se obtuvo una exactitud del 97.46%, en el entrenamiento solo fue necesario 18 épocas, con 500 pasos por época, tomando un aproximado de 5 horas y media.

Épocas	Exactitud		Error medio cuad(EMC)		Pérdida(loss)	
	Validación	Entrenamiento	Validación	Entrenam.	Loss-Val.	Loss-Entre.
1	0.30094	0.08184	0.00687	0.00798	3.18583	4.09086
2	0.60844	0.36915	0.00457	0.00638	1.55808	2.29750
3	0.75969	0.59967	0.00275	0.00442	1.38904	1.36475
4	0.81313	0.73705	0.00222	0.00302	0.50591	0.86047
5	0.89063	0.79927	0.00138	0.00237	0.41317	0.65440
6	0.88719	0.83062	0.00137	0.00197	0.34757	0.53414
7	0.90406	0.85959	0.00114	0.00168	0.02183	0.44187
8	0.90125	0.87470	0.00118	0.00152	0.09785	0.40100
9	0.93281	0.88370	0.00088	0.00141	0.10390	0.36971
10	0.91625	0.89967	0.00106	0.00118	0.23731	0.30646
11	0.92781	0.90343	0.00089	0.00118	0.36427	0.30390
12	0.96812	0.94246	0.00041	0.00071	0.00519	0.18370
13	0.96938	0.96096	0.00039	0.00049	0.07342	0.12494
14	0.97031	0.96710	0.00038	0.00043	0.06063	0.11487
15	0.97031	0.96246	0.00034	0.00046	0.01753	0.11609
16	0.97250	0.96884	0.00034	0.00038	0.13049	0.09661
17	0.97500	0.97135	0.00030	0.00035	0.02172	0.08904
18	0.97469	0.97536	0.00030	0.00033	0.05559	0.08335

Tabla 14. Datos de entrenamiento algoritmo CN-GC, base de firmas DBGC.

Los resultados del entrenamiento del algoritmo CN-GC para la base de datos DB-GC se indican en la tabla 14, fue suficiente 18 épocas, con 500 pasos por cada época, a partir de la época 18 el algoritmo ya no disminuye su error. Se usó el método EarlyStopping, proporcionado por la librería Keras para optimizar el tiempo de entrenamiento. El tiempo de entrenamiento necesario fue de aproximadamente cinco horas y media. Gráficamente la figura 49 muestra la exactitud del entrenamiento, la figura 50 el error medio cuadrático y la figura 51 los valores de la función de pérdida, estos datos fueron tomados de la tabla 14.

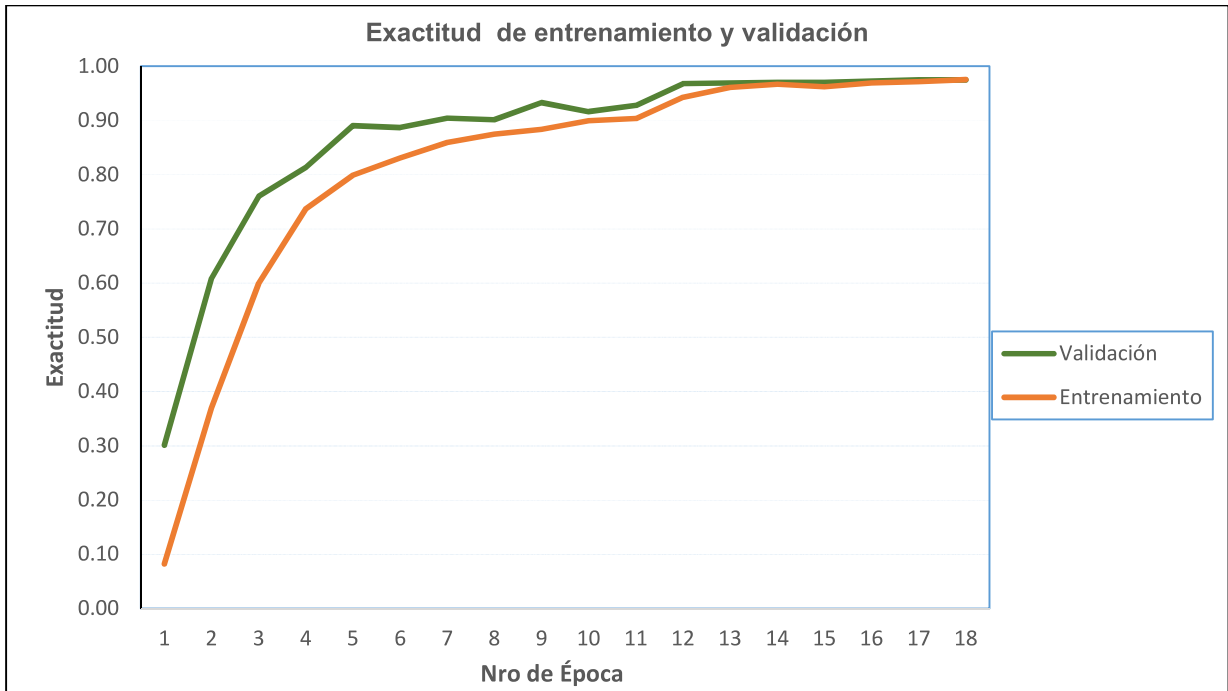


Figura 49. Exactitud de validación y entrenamiento, algoritmo CN-GC, base DB-GC.

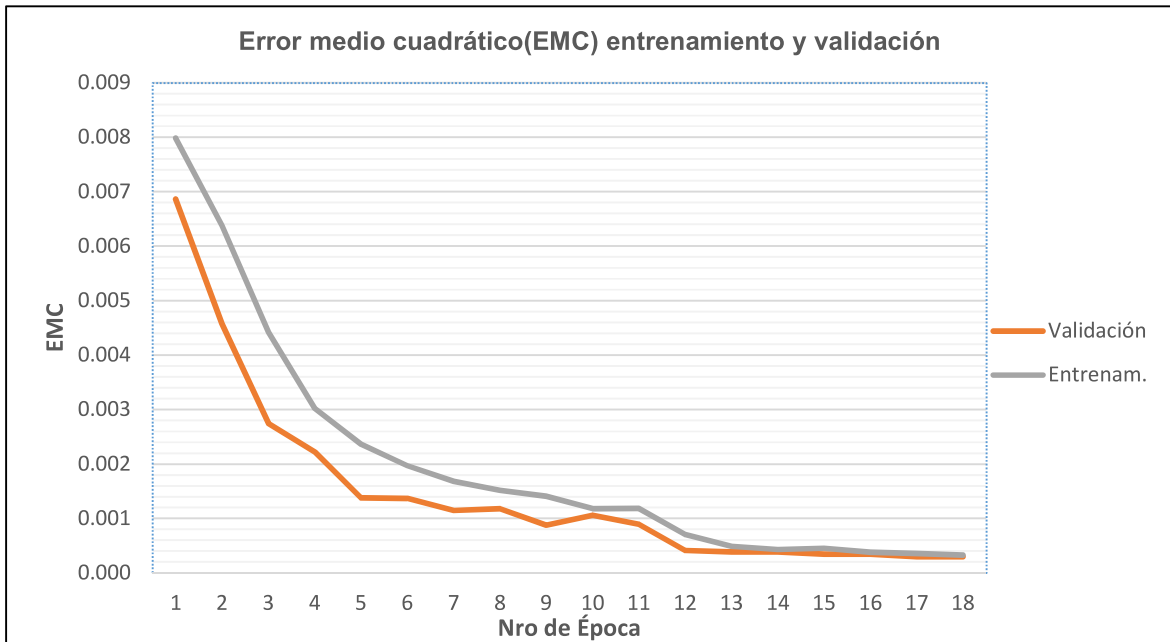


Figura 50. EMC de validación y entrenamiento, algoritmo CN-GC, base DB-GC.

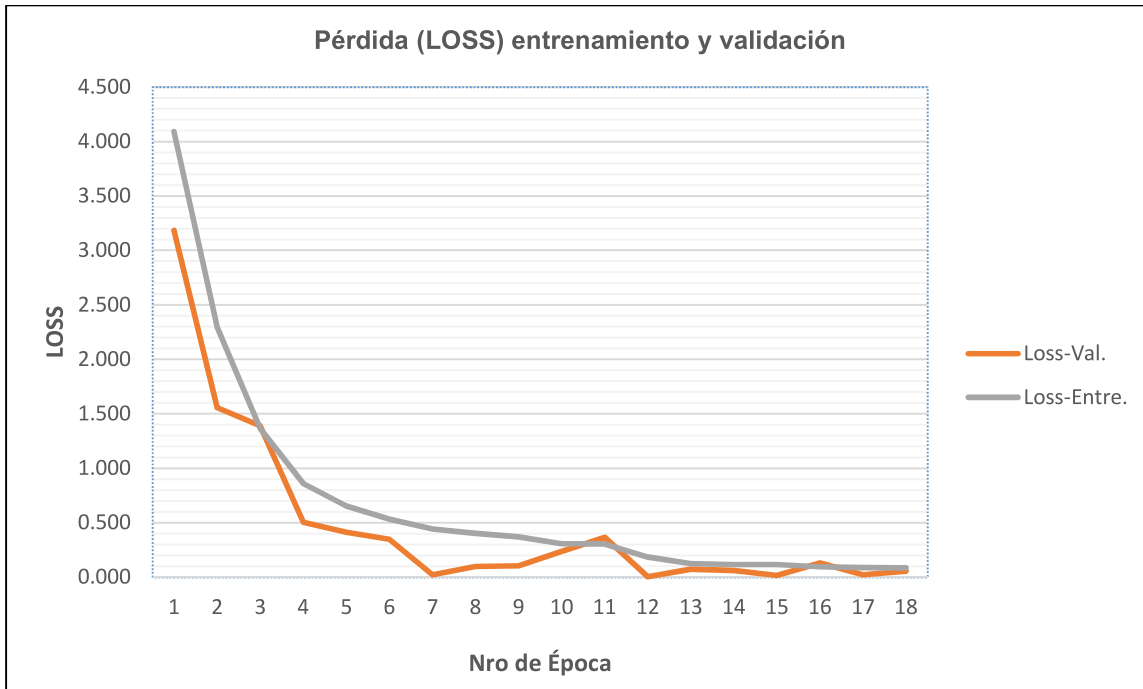


Figura 51. LOSS de validación y entrenamiento, algoritmo CN-GC, base DB-GC.

Luego de entrenar el modelo con la base de datos DB-GC, se procedió a ejecutar el algoritmo con datos de prueba, los resultados se presentan en la tabla 15. Se puede observar que en esta fase se logró alcanzar una exactitud de **91.38%**, mientras que en el entrenamiento se alcanzó un **97.46%** con la base de datos de firmas DB-GC.

MODELO CN-GC, BASE DB-GC						F1-Score
CLASES	PRES	SEN	ESP	TFP		
1	1	0.875	1	0		0.9333
2	1	1	1	0		1
3	1	1	1	0		1
4	0.8571	0.75	0.9988	0.0012		0.8
5	1	0.875	1	0		0.9333
6	1	0.875	1	0		0.9333
7	1	0.875	1	0		0.9333
8	1	1	1	0		1
9	1	0.875	1	0		0.9333
10	1	1	1	0		1
11	0.8571	0.8571	0.9988	0.0012		0.8571
12	0.7143	1	0.9976	0.0024		0.8333
13	1	1	1	0		1
14	1	0.875	1	0		0.9333
15	0.8571	0.5455	0.9988	0.0012		0.6667
16	1	0.875	1	0		0.9333
17	0.8571	0.75	0.9988	0.0012		0.8

18	0.7143	1	0.9976	0.0024	0.8333
19	0.1429	1	0.9929	0.0071	0.2501
20	0.7143	1	0.9976	0.0024	0.8333
21	1	1	1	0	1
22	0.4286	1	0.9953	0.0047	0.6
23	1	1	1	0	1
24	1	1	1	0	1
25	0.7143	1	0.9976	0.0024	0.8333
26	1	0.875	1	0	0.9333
27	0.4286	0.4286	0.9952	0.0048	0.4286
28	1	1	1	0	1
29	0.8571	0.8571	0.9988	0.0012	0.8571
30	0.8571	1	0.9988	0.0012	0.9231
31	1	1	1	0	1
32	1	0.875	1	0	0.9333
33	1	1	1	0	1
34	1	1	1	0	1
35	1	0.7778	1	0	0.875
36	1	1	1	0	1
37	0.5714	1	0.9964	0.0036	0.7272
38	1	1	1	0	1
39	1	1	1	0	1
40	1	1	1	0	1
41	0.4286	1	0.9953	0.0047	0.6
42	1	1	1	0	1
43	1	0.875	1	0	0.9333
44	1	1	1	0	1
45	1	1	1	0	1
46	1	0.5385	1	0	0.7
47	1	0.7778	1	0	0.875
48	1	1	1	0	1
49	1	1	1	0	1
50	1	1	1	0	1
51	1	1	1	0	1
52	1	1	1	0	1
53	1	1	1	0	1
54	1	1	1	0	1
55	1	1	1	0	1
56	1	0.875	1	0	0.9333
57	1	0.7778	1	0	0.875
58	1	1	1	0	1
59	1	1	1	0	1
60	0.1429	1	0.9929	0.0071	0.2501
61	1	1	1	0	1
62	0.7143	0.8333	0.9976	0.0024	0.7692
63	1	0.7778	1	0	0.875
64	1	1	1	0	1
65	0.8571	1	0.9988	0.0012	0.9231
66	1	1	1	0	1
67	1	1	1	0	1
68	0.5714	1	0.9964	0.0036	0.7272
69	1	1	1	0	1
70	0.8571	1	0.9988	0.0012	0.9231
71	1	1	1	0	1
72	0.4286	0.3333	0.9952	0.0048	0.375
73	0.8571	1	0.9988	0.0012	0.9231

74	1	1	1	0	1
75	1	0.875	1	0	0.9333
76	1	1	1	0	1
77	0.8571	1	0.9988	0.0012	0.9231
78	1	1	1	0	1
79	1	0.875	1	0	0.9333
80	1	1	1	0	1
81	1	1	1	0	1
82	1	1	1	0	1
83	1	1	1	0	1
84	1	1	1	0	1
85	1	0.7778	1	0	0.875
86	1	1	1	0	1
87	1	1	1	0	1
88	1	0.875	1	0	0.9333
89	1	0.875	1	0	0.9333
90	1	1	1	0	1
91	0.8571	1	0.9988	0.0012	0.9231
92	1	1	1	0	1
93	0.8571	1	0.9988	0.0012	0.9231
94	1	1	1	0	1
95	1	1	1	0	1
96	1	1	1	0	1
97	1	1	1	0	1
98	1	0.7778	1	0	0.875
99	0.4286	1	0.9953	0.0047	0.6
100	1	1	1	0	1
101	1	1	1	0	1
102	0.5714	1	0.9964	0.0036	0.7272
103	1	0.875	1	0	0.9333
104	1	1	1	0	1
105	1	1	1	0	1
106	1	1	1	0	1
107	0.4286	1	0.9953	0.0047	0.6
108	1	0.7778	1	0	0.875
109	1	1	1	0	1
110	1	0.7	1	0	0.8235
111	1	0.5833	1	0	0.7368
112	0.7143	1	0.9976	0.0024	0.8333
113	1	0.5385	1	0	0.7
114	1	0.875	1	0	0.9333
115	1	1	1	0	1
116	0.5714	1	0.9964	0.0036	0.7272
117	1	1	1	0	1
118	1	1	1	0	1
119	1	1	1	0	1
120	0.8571	1	0.9988	0.0012	0.9231
121	1	1	1	0	1
Exact	0.9138	0	0	0	0
Prom	0.91381	0.93417	0.99928	0.00072	0.90798
DesvSt	0.18226	0.12687	0.00152	0.00152	0.14915

Tabla 15. Datos fase de validación, algoritmo CN-GC, base DB-GC.

A continuación, En la figura 52, se muestra la matriz de confusión para el dataset DB-GC.

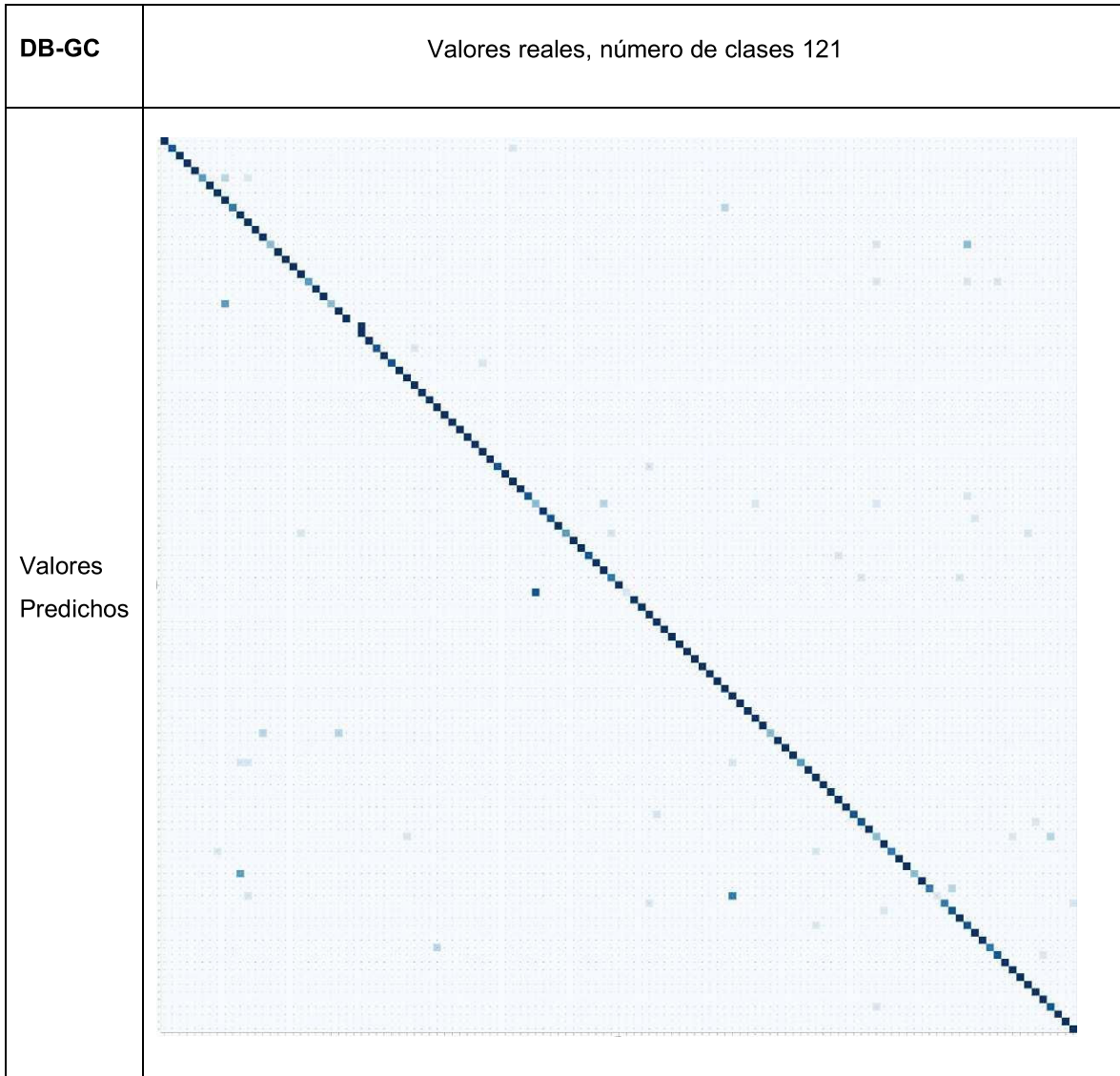


Figura 52. Matriz de confusión del algoritmo CN-GC, base de datos DB-GC.

En la figura 52, se muestra la matriz de confusión para la base de datos DB-GC que tiene 121 clases. Cada celda pintada de azul oscuro en la diagonal indica una validación al 100%, las celdas que están fuera de la diagonal, pintados de un color azul claro, indican que en esa clase produjo una validación inferior al 99%. Se puede apreciar que existen problemas en 13 clases aproximadamente, en las demás clases, la predicción fue correcta.

4.1.3. Entrenamiento del algoritmo Dígitos, base CEDAR.

Este algoritmo se procede a entrenar usando la base CEDAR, se logró una exactitud de **95.19%**, siendo necesarias 26 épocas y 400 pasos en cada época. Sin embargo, en la fase de validación su exactitud bajo a **82,73%** para este modelo fue necesario aproximadamente tres horas. Se puede observar que la diferencia es muy grande y superior al 10%, lo que indica un probable overfitting. El algoritmo está orientado al reconocimiento de dígitos manuscritos, al implementar el algoritmo fue necesario hacer un cambio, aumentando una capa inicial de dimensión 340x440, como se puede apreciar en la tabla 6, debido a que la dimensión de la imagen original es de 28x28, razón por la que el algoritmo pudo no acoplarse eficientemente. La figura 53 muestra la exactitud, la figura 54 el error medio cuadrático y la figura 55 los valores de la función de pérdida, estos datos fueron tomados de la tabla 16.

Épocas	Exactitud		Error medio cuadrático(EMC)		Pérdida(loss)	
	Validación	Entrenam.	Validación	Entrenam.	Loss-Val.	Loss-Entre.
1	0.26104	0.06053	0.01579	0.01758	2.93143	3.77948
2	0.52339	0.24284	0.01094	0.01552	1.90910	2.72125
3	0.72293	0.43064	0.00738	0.01271	1.33163	1.98550
4	0.78606	0.56240	0.00583	0.01028	0.90238	1.51629
5	0.81158	0.66467	0.00451	0.00824	0.48588	1.14903
6	0.86981	0.71091	0.00365	0.00718	0.61532	0.96195
7	0.86915	0.76419	0.00326	0.00588	0.51690	0.76121
8	0.89009	0.80422	0.00294	0.00498	1.14294	0.63060
9	0.89849	0.81936	0.00280	0.00458	0.05600	0.57300
10	0.92051	0.84237	0.00238	0.00407	0.38546	0.51251
11	0.91724	0.84899	0.00237	0.00396	0.60407	0.49718
12	0.90383	0.87071	0.00247	0.00341	0.02939	0.41468
13	0.91102	0.88241	0.00238	0.00301	0.67687	0.37008
14	0.92607	0.88020	0.00201	0.00303	0.00575	0.38878
15	0.93327	0.89494	0.00169	0.00280	0.25947	0.34251
16	0.93523	0.90290	0.00184	0.00260	0.01810	0.31624
17	0.92673	0.91236	0.00193	0.00237	0.04500	0.28962
18	0.93495	0.91709	0.00161	0.00220	0.00000	0.26329
19	0.94570	0.91866	0.00151	0.00217	0.29563	0.25023
20	0.94504	0.92907	0.00145	0.00187	0.00428	0.21370
21	0.93490	0.92308	0.00167	0.00201	0.13632	0.23624
22	0.94897	0.92891	0.00154	0.00193	0.57760	0.22302
23	0.94962	0.94629	0.00135	0.00149	0.05201	0.17109
24	0.94897	0.95050	0.00131	0.00135	0.34955	0.16182
25	0.95289	0.95156	0.00125	0.00139	0.43767	0.16086
26	0.95191	0.95397	0.00128	0.00126	0.02875	0.15597

Tabla 16. Datos de entrenamiento, CNN reconocimiento dígitos, base CEDAR.

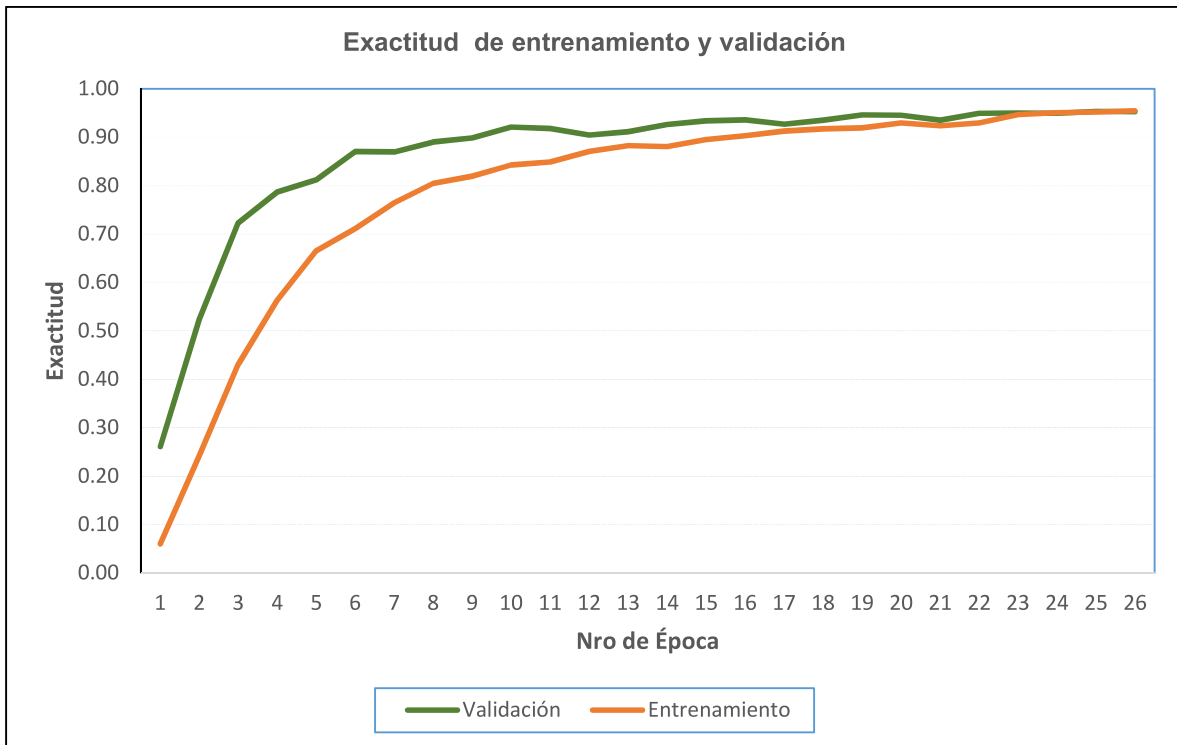


Figura 53. Exactitud de entrenamiento, CNN reconocimiento de dígitos, base CEDAR.

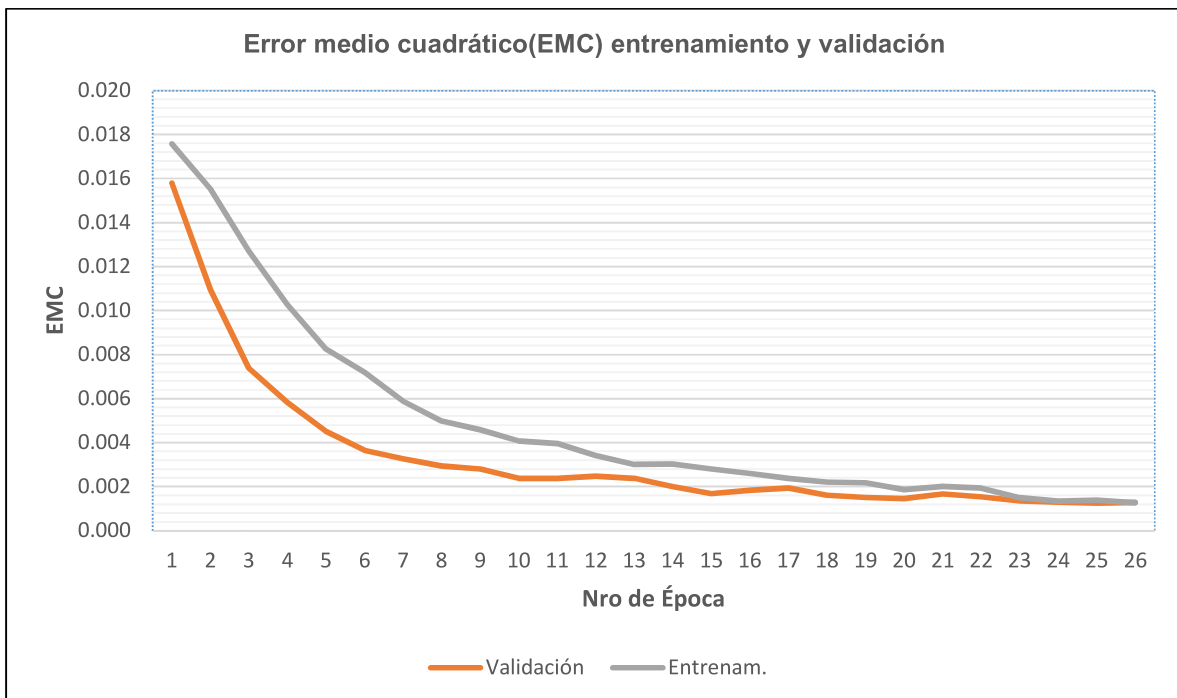


Figura 54. EMC de entrenamiento, CNN reconocimiento de dígitos, base CEDAR

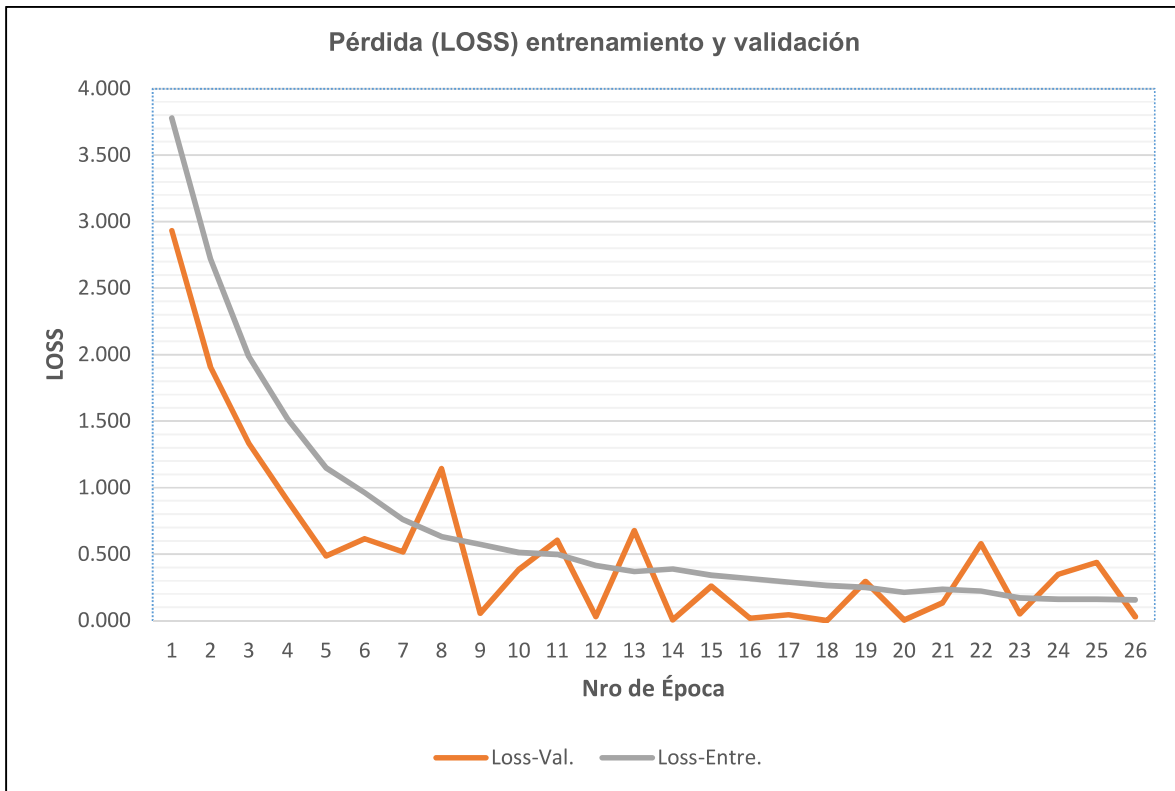


Figura 55, LOSS de entrenamiento, CNN reconocimiento de dígitos, base CEDAR.

A continuación, los datos de la fase de validación del segundo algoritmo para la base de datos CEDAR, se muestra la tabla de datos y luego su matriz de confusión. En esta etapa la exactitud que se alcanzó con el algoritmo de Dígitos Manuscritos fue de **82.73%**.

MODELO DIGITOS CEDAR					
CLASES	PRES	SEN	ESP	TFP	F1
1	1	1	1	0	1
2	0.75	0.6	0.9953	0.0047	0.6667
3	1	0.8	1	0	0.8889
4	1	1	1	0	1
5	0.75	0.75	0.9954	0.0046	0.75
6	1	0.8	1	0	0.8889
7	1	1	1	0	1
8	1	1	1	0	1
9	1	0.8	1	0	0.8889
10	0.75	1	0.9954	0.0046	0.8571
11	0.75	1	0.9954	0.0046	0.8571
12	1	1	1	0	1
13	1	0.8	1	0	0.8889

14	1	0.8	1	0	0.8889
15	1	1	1	0	1
16	1	1	1	0	1
17	1	0.8	1	0	0.8889
18	0.5	1	0.9908	0.0092	0.6667
19	0.75	1	0.9954	0.0046	0.8571
20	0.75	0.375	0.9953	0.0047	0.5
21	0.75	1	0.9954	0.0046	0.8571
22	0.5	0.6667	0.9908	0.0092	0.5714
23	1	0.6667	1	0	0.8
24	0.75	1	0.9954	0.0046	0.8571
25	0	0	0.9818	0.0182	0
26	0.75	0.75	0.9954	0.0046	0.75
27	0.75	1	0.9954	0.0046	0.8571
28	1	0.8	1	0	0.8889
29	1	1	1	0	1
30	0.75	1	0.9954	0.0046	0.8571
31	1	1	1	0	1
32	0.5	0.6667	0.9908	0.0092	0.5714
33	0	0	0.9817	0.0183	0
34	1	0.8	1	0	0.8889
35	0.5	1	0.9908	0.0092	0.6667
36	0.5	1	0.9908	0.0092	0.6667
37	0.75	1	0.9954	0.0046	0.8571
38	0.75	1	0.9954	0.0046	0.8571
39	1	0.8	1	0	0.8889
40	0.5	0.4	0.9907	0.0093	0.4444
41	1	0.8	1	0	0.8889
42	0.75	1	0.9954	0.0046	0.8571
43	1	0.8	1	0	0.8889
44	0.75	1	0.9954	0.0046	0.8571
45	1	1	1	0	1
46	1	0.8	1	0	0.8889
47	1	0.5714	1	0	0.7272
48	1	1	1	0	1
49	1	1	1	0	1
50	1	0.8	1	0	0.8889
51	0.25	1	0.9863	0.0137	0.4
52	1	1	1	0	1
53	1	0.6667	1	0	0.8
54	1	0.6667	1	0	0.8
55	1	0.8	1	0	0.8889
Exact	0.8273	0	0	0	0
Prom	0.8273	0.836	0.9968	0.0032	0.8119
DesvSt	0.2471	0.2279	0.0045	0.0045	0.2132

Tabla 17, Resultados fase de validación, segundo algoritmo, base de datos CEDAR.

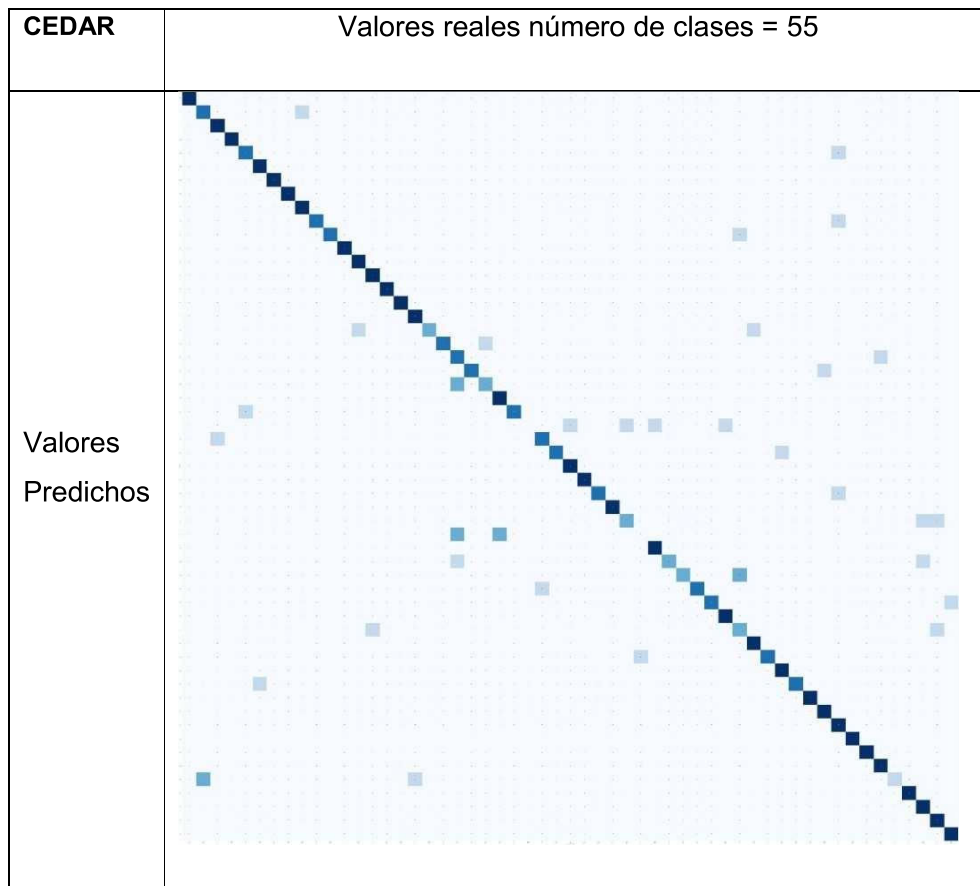


Figura 56. Matriz de confusión, algoritmo reconocimiento de dígitos, base CEDAR.

En la figura 56, se muestra la matriz de confusión para el dataset CEDAR para el algoritmo reconocimiento de dígitos, este dataset, tiene 55 clases. Cada celda pintada de azul oscuro en la diagonal indica una validación al 100%, las celdas que están fuera de la diagonal, indican que en esa clase produjo una validación inferior al 99%.

4.1.4. Entrenamiento algoritmo Dígitos, base DB-GC.

Los datos de entrenamiento del algoritmo de reconocimiento de Dígitos, muestran que fue necesario 27 épocas, con 400 pasos por época para la base DB-GC, alcanzando una exactitud de **82.26%** (tabla18), en el entrenamiento y de **76.62%**, en la validación como muestra la tabla 19. Para entrenar este modelo fue necesario aproximadamente cuatro horas.

Épocas	Error medio cuadrático					
	Exactitud		(EMC)		Pérdida(loss)	
	Validación	Entrenamiento	Validación	Entrenam.	Loss-Val.	Loss-Ent
1	0.0571	0.0150	0.0081	0.0082	4.4077	4.7120
2	0.1639	0.0754	0.0077	0.0080	3.9392	4.0573
3	0.2954	0.1351	0.0070	0.0078	4.0574	3.5404
4	0.3556	0.2171	0.0065	0.0074	3.3282	3.1057
5	0.4350	0.2799	0.0060	0.0071	2.7591	2.8099
6	0.5055	0.3281	0.0052	0.0067	2.8638	2.5115
7	0.6130	0.3732	0.0047	0.0064	1.1870	2.3091
8	0.6238	0.4426	0.0045	0.0058	1.4062	2.0421
9	0.6720	0.4751	0.0039	0.0055	1.5639	1.8768
10	0.6279	0.5100	0.0040	0.0052	1.8458	1.7136
11	0.7128	0.5260	0.0033	0.0050	1.0908	1.6730
12	0.7189	0.5565	0.0033	0.0048	0.4589	1.5392
13	0.7442	0.5849	0.0031	0.0045	1.6083	1.4408
14	0.7376	0.6119	0.0030	0.0042	1.3137	1.3081
15	0.7515	0.6200	0.0028	0.0042	1.2943	1.2737
16	0.7775	0.6428	0.0026	0.0039	0.7011	1.1846
17	0.8117	0.6738	0.0022	0.0037	1.5443	1.1205
18	0.8207	0.6950	0.0022	0.0034	0.5826	1.0154
19	0.8298	0.6816	0.0021	0.0035	0.2683	1.0262
20	0.8235	0.6913	0.0021	0.0035	0.5400	1.0264
21	0.8279	0.7031	0.0021	0.0033	0.9518	0.9727
22	0.8288	0.7163	0.0020	0.0032	0.2858	0.9390
23	0.8293	0.7076	0.0020	0.0032	0.4869	0.9224
24	0.8241	0.7169	0.0020	0.0032	0.6032	0.9483
25	0.8260	0.7047	0.0021	0.0032	1.7119	0.9302
26	0.8241	0.7201	0.0020	0.0032	1.6380	0.9287
27	0.8226	0.7232	0.0020	0.0031	2.0238	0.9031

Tabla 18. Datos entrenamiento, algoritmo dos, base DB-GC.

Los datos de la tabla 18, se muestran de manera gráfica en las figuras 57, 58, 59. En la figura 57 se muestra la exactitud para el algoritmo reconocimiento de dígitos para el dataset DB-GC, la figura 58 muestra el error medio cuadrático y en la figura 59 se muestra los valores de la función de pérdida.

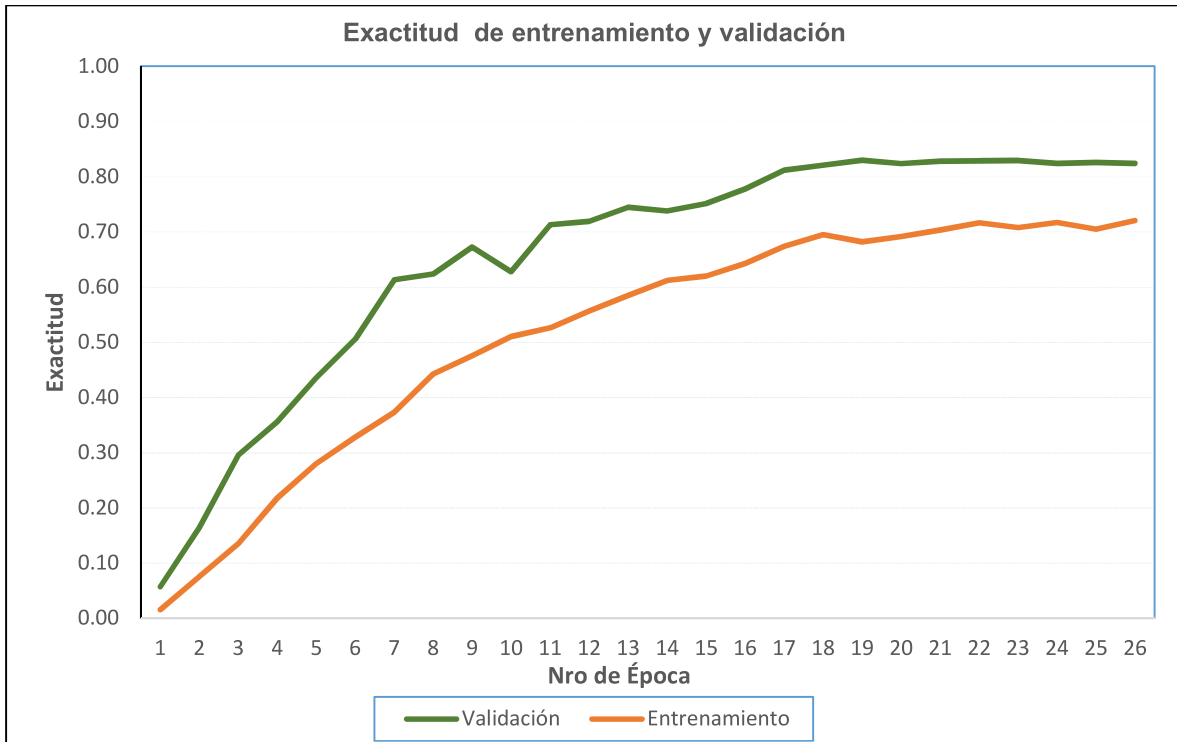


Figura 57. Exactitud de entrenamiento, segundo algoritmo, base DB-GC.

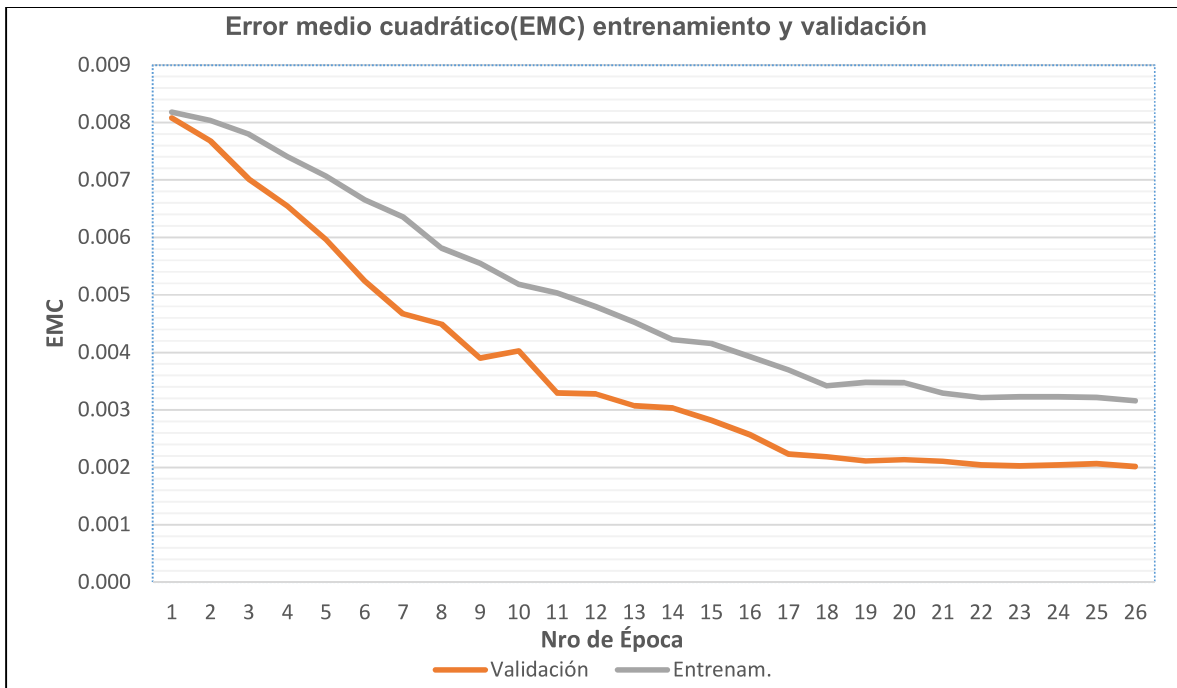


Figura 58, EMC de entrenamiento, CNN reconocimiento de dígitos, base DB-GC.

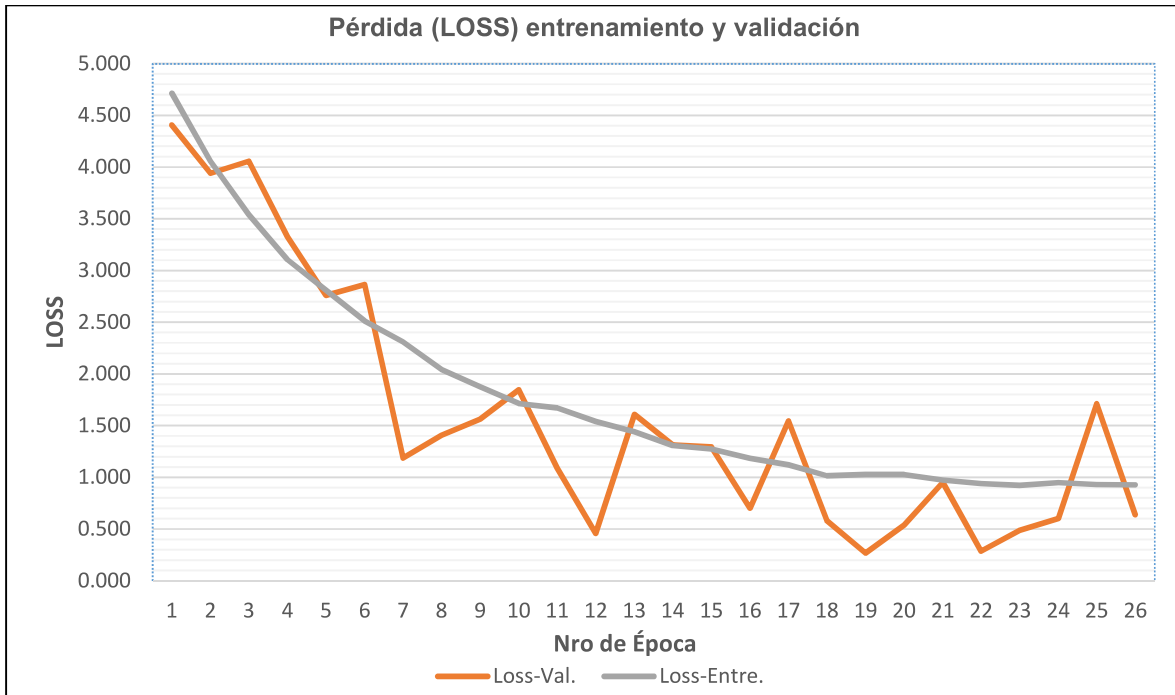


Figura 59. LOSS de entrenamiento, CNN reconocimiento de dígitos,, base DB-GC.

En la fase de validación con el algoritmo de reconocimiento de dígitos, se alcanzó una exactitud del **76.62%** (tabla19), en la etapa de entrenamiento fue de **82.26%**.

RECONOCIMIENTO DE DIGITOS DB-GC						F1-
CLASES	PRES	SEN	ESP	TFP	Score	
1	0.8571	1	0.9988	0.0012	0.9231	
2	1	0.7778	1	0	0.875	
3	0.7143	0.8333	0.9976	0.0024	0.7692	
4	0.8571	0.8571	0.9988	0.0012	0.8571	
5	0.8571	1	0.9988	0.0012	0.9231	
6	1	0.875	1	0	0.9333	
7	1	0.5833	1	0	0.7368	
8	0.8571	0.75	0.9988	0.0012	0.8	
9	0.4286	0.5	0.9952	0.0048	0.4616	
10	0.5714	0.6667	0.9964	0.0036	0.6154	
11	0.7143	0.625	0.9976	0.0024	0.6667	
12	0	0	0.9917	0.0083	0	
13	0.4286	1	0.9953	0.0047	0.6	
14	0.5714	1	0.9964	0.0036	0.7272	
15	0.2857	0.6667	0.9941	0.0059	0.4	
16	0.8571	0.5455	0.9988	0.0012	0.6667	
17	0.8571	0.75	0.9988	0.0012	0.8	
18	0.5714	1	0.9964	0.0036	0.7272	
19	0.2857	0.5	0.9941	0.0059	0.3636	
20	0.5714	0.8	0.9964	0.0036	0.6666	
21	0.5714	1	0.9964	0.0036	0.7272	
22	0	0	0.9917	0.0083	0	
23	0.5714	1	0.9964	0.0036	0.7272	

24	0.8571	0.8571	0.9988	0.0012	0.8571
25	0.8571	0.8571	0.9988	0.0012	0.8571
26	0.2857	0.5	0.9941	0.0059	0.3636
27	0	0	0.9917	0.0083	0
28	0.8571	1	0.9988	0.0012	0.9231
29	0.2857	0.3333	0.9941	0.0059	0.3077
30	0.7143	1	0.9976	0.0024	0.8333
31	0.4286	0.75	0.9953	0.0047	0.5455
32	0.1429	0.3333	0.9929	0.0071	0.2
33	0.5714	0.4	0.9964	0.0036	0.4706
34	1	0.7778	1	0	0.875
35	0.5714	0.8	0.9964	0.0036	0.6666
36	0.8571	0.75	0.9988	0.0012	0.8
37	0.2857	1	0.9941	0.0059	0.4444
38	0.2857	0.4	0.9941	0.0059	0.3333
39	1	0.7	1	0	0.8235
40	1	0.5833	1	0	0.7368
41	0	0	0.9917	0.0083	0
42	1	0.875	1	0	0.9333
43	0.8571	0.75	0.9988	0.0012	0.8
44	0.4286	1	0.9953	0.0047	0.6
45	1	0.7778	1	0	0.875
46	0.7143	0.8333	0.9976	0.0024	0.7692
47	1	0.875	1	0	0.9333
48	0.4286	0.75	0.9953	0.0047	0.5455
49	1	0.875	1	0	0.9333
50	1	1	1	0	1
51	1	1	1	0	1
52	0.8571	1	0.9988	0.0012	0.9231
53	1	0.7	1	0	0.8235
54	1	0.6364	1	0	0.7778
55	0.8571	1	0.9988	0.0012	0.9231
56	0.8571	0.8571	0.9988	0.0012	0.8571
57	1	1	1	0	1
58	1	0.875	1	0	0.9333
59	1	0.875	1	0	0.9333
60	0.7143	1	0.9976	0.0024	0.8333
61	0.7143	0.8333	0.9976	0.0024	0.7692
62	1	0.875	1	0	0.9333
63	0.5714	0.4444	0.9964	0.0036	0.5
64	0.7143	1	0.9976	0.0024	0.8333
65	0.5714	0.4444	0.9964	0.0036	0.5
66	1	0.7	1	0	0.8235
67	0.5714	0.5714	0.9964	0.0036	0.5714
68	0.8571	0.6667	0.9988	0.0012	0.75
69	0.7143	0.8333	0.9976	0.0024	0.7692
70	0.4286	1	0.9953	0.0047	0.6
71	0.7143	0.8333	0.9976	0.0024	0.7692
72	0.8571	0.6667	0.9988	0.0012	0.75
73	0.8571	0.6	0.9988	0.0012	0.7059
74	1	1	1	0	1
75	1	1	1	0	1
76	1	1	1	0	1
77	1	0.875	1	0	0.9333
78	1	1	1	0	1
79	0.8571	1	0.9988	0.0012	0.9231

80	1	0.875	1	0	0.9333
81	1	1	1	0	1
82	1	0.875	1	0	0.9333
83	1	1	1	0	1
84	1	0.5385	1	0	0.7
85	0.5714	0.6667	0.9964	0.0036	0.6154
86	1	0.875	1	0	0.9333
87	1	0.7778	1	0	0.875
88	0.4286	0.75	0.9953	0.0047	0.5455
89	1	0.875	1	0	0.9333
90	1	0.7778	1	0	0.875
91	0.5714	1	0.9964	0.0036	0.7272
92	1	0.875	1	0	0.9333
93	0.8571	0.8571	0.9988	0.0012	0.8571
94	1	0.7	1	0	0.8235
95	1	0.7778	1	0	0.875
96	0.8571	0.6667	0.9988	0.0012	0.75
97	0.4286	0.75	0.9953	0.0047	0.5455
98	0.5714	0.3077	0.9964	0.0036	0.4
99	1	0.7778	1	0	0.875
100	0.8571	0.4615	0.9988	0.0012	0.6
101	1	0.875	1	0	0.9333
102	1	1	1	0	1
103	0.8571	0.75	0.9988	0.0012	0.8
104	1	0.875	1	0	0.9333
105	1	0.875	1	0	0.9333
106	0.8571	0.75	0.9988	0.0012	0.8
107	0.5714	0.8	0.9964	0.0036	0.6666
108	0.5714	0.6667	0.9964	0.0036	0.6154
109	0.8571	0.6667	0.9988	0.0012	0.75
110	1	1	1	0	1
111	0.8571	0.75	0.9988	0.0012	0.8
112	1	1	1	0	1
113	0.8571	0.6	0.9988	0.0012	0.7059
114	0.8571	1	0.9988	0.0012	0.9231
115	0.5714	1	0.9964	0.0036	0.7272
116	0.7143	1	0.9976	0.0024	0.8333
117	0.7143	0.8333	0.9976	0.0024	0.7692
118	0.7143	0.625	0.9976	0.0024	0.6667
119	1	1	1	0	1
120	1	0.875	1	0	0.9333
121	1	0.7778	1	0	0.875
Exact	0.7662	0	0	0	0
Prom	0.76622	0.77437	0.99805	0.00195	0.75146
DesvSt	0.26424	0.22737	0.00219	0.00219	0.22426

Tabla 19. Resultados fase de prueba, algoritmo Dígitos, base de datos DB-GC.

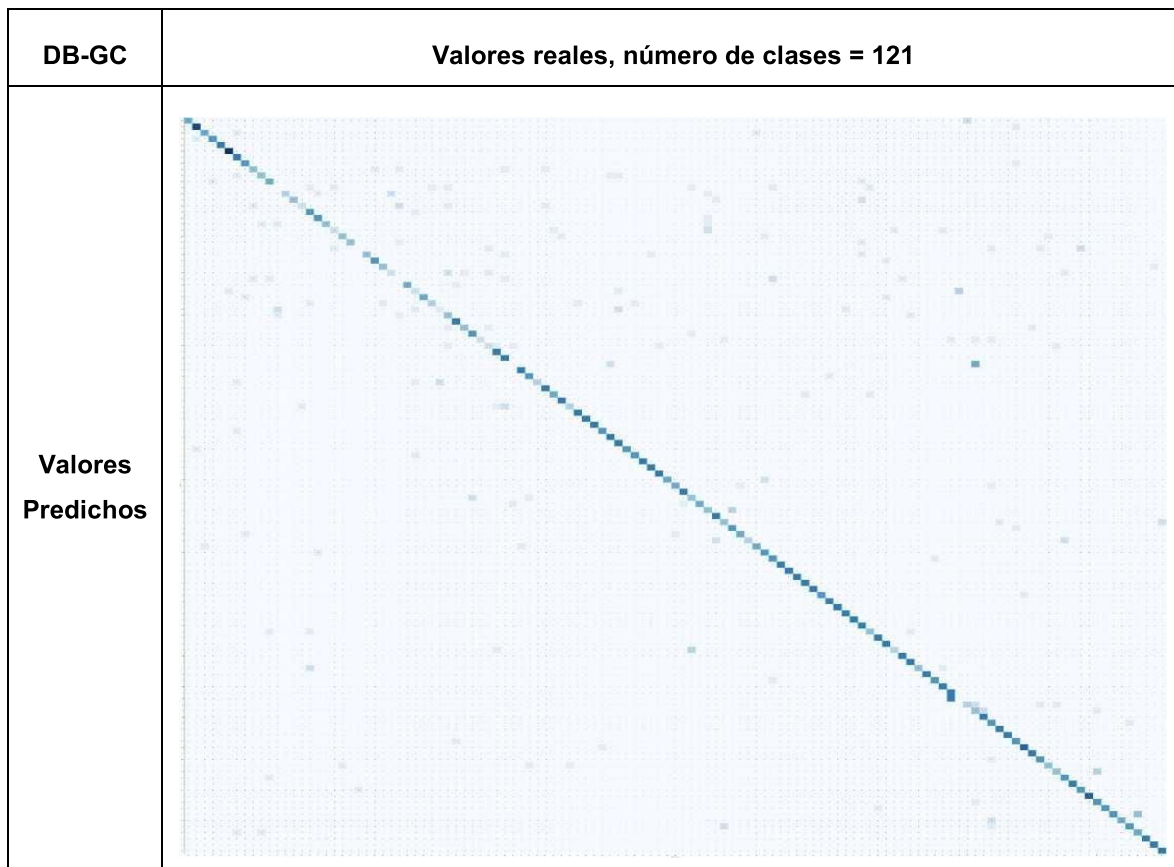


Figura 60. Matriz de confusión, algoritmo Dígitos, base DB-GC.

En la figura 60, se muestra la matriz de confusión para el dataset DB-GC para el algoritmo reconocimiento de dígitos, este dataset, tiene 121 clases. Cada celda pintada de azul oscuro en la diagonal indica una validación al 100%, las celdas que están fuera de la diagonal, indican que en esa clase produjo una validación inferior al 99%.

4.1.5. Entrenamiento red convolucional poco profunda, base CEDAR.

Este algoritmo tiene muy pocas capas de convolución, no es muy complicado en la capa de extracción de características y en la capa de clasificación, de ahí su nombre Red Convolutiva Poco Profunda, es un algoritmo muy liviano y muy efectivo [37], en el trabajo de investigación referenciado se puede encontrar los datos como una evidencia, se debe mencionar que solo fue suficiente 15 épocas de entrenamiento para la base CEDAR y que fue necesario una hora y media aproximadamente para completar su entrenamiento. En la tabla 20, se muestran los datos de la fase entrenamiento para esta red convolucional.

Épocas	Exactitud		Error medio cuadrático(EMC)		Pérdida(loss)	
	Validación	Entrenam.	Validación	Entrenam.	Loss-Val.	Loss-Entre.
1	0.5532	0.1200	0.0119	0.0170	2.1483	3.5230
2	0.8557	0.5908	0.0037	0.0099	0.3808	1.4795
3	0.9369	0.7967	0.0018	0.0053	0.1369	0.6938
4	0.9444	0.8670	0.0014	0.0036	0.1567	0.4649
5	0.9428	0.9016	0.0016	0.0026	0.2120	0.3304
6	0.9607	0.9228	0.0011	0.0021	0.0198	0.2534
7	0.9778	0.9402	0.0007	0.0016	0.5171	0.1992
8	0.9562	0.9377	0.0011	0.0017	0.1347	0.2145
9	0.9593	0.9466	0.0011	0.0015	0.0000	0.1822
10	0.9709	0.9554	0.0008	0.0012	0.0821	0.1395
11	0.9709	0.9600	0.0008	0.0011	0.1216	0.1352
12	0.9771	0.9606	0.0006	0.0011	0.0011	0.1376
13	0.9787	0.9619	0.0006	0.0010	0.0219	0.1166
14	0.9859	0.9841	0.0004	0.0004	0.0672	0.0520
15	0.9836	0.9850	0.0005	0.0004	0.0008	0.0487

Tabla 20. Datos de entrenamiento, algoritmo poco profundo, base CEDAR.

Los datos de la tabla 20, se muestran de manera gráfica en las figuras 61, 62, 63. En la figura 61 se muestra la exactitud para el algoritmo reconocimiento de dígitos para el dataset DB-GC, la figura 61 muestra el error medio cuadrático y en la figura 63 se muestra los valores de la función de pérdida.

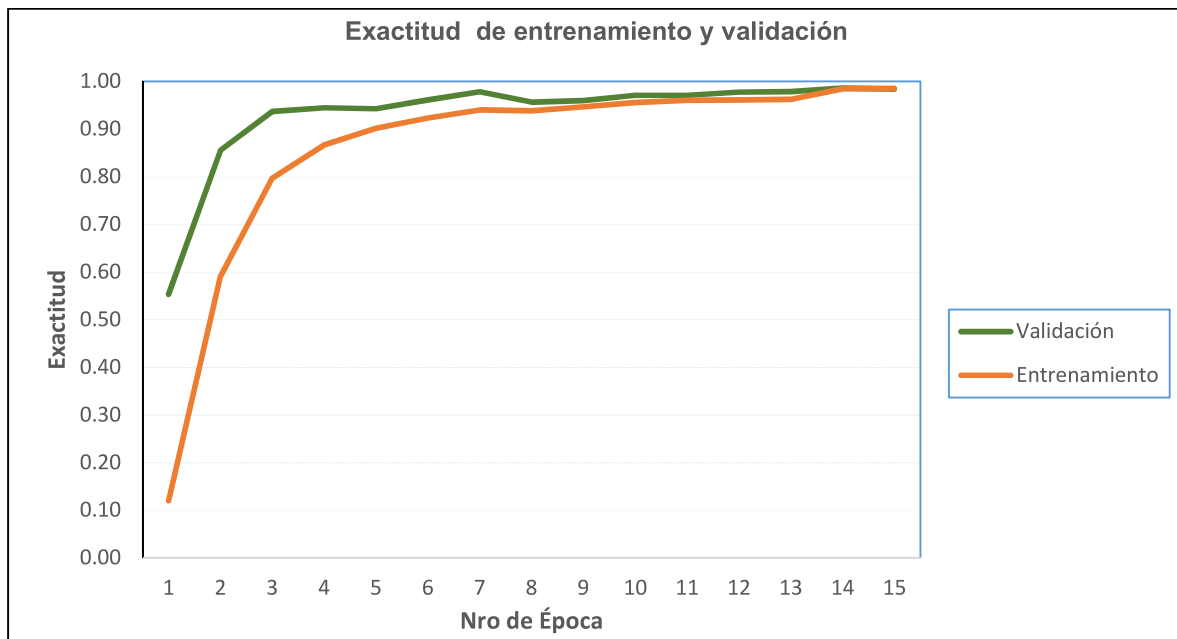


Figura 61. Exactitud de entrenamiento, tercer algoritmo, base CEDAR

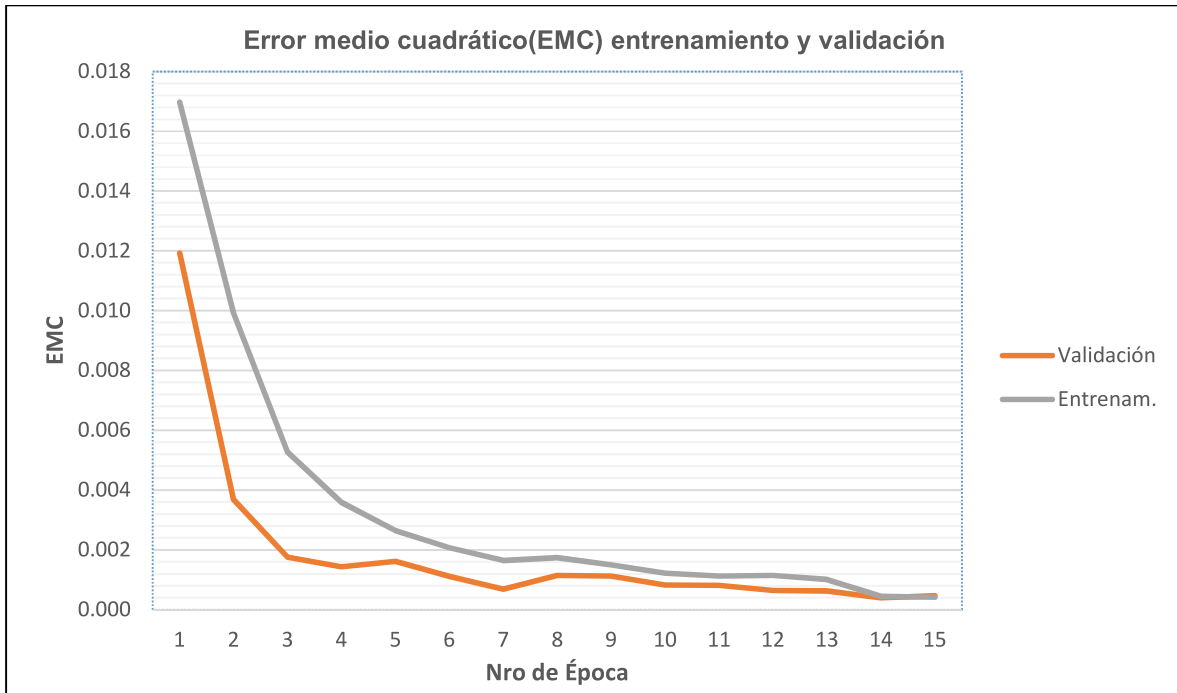


Figura 62. EMC de entrenamiento, tercer algoritmo, base CEDAR

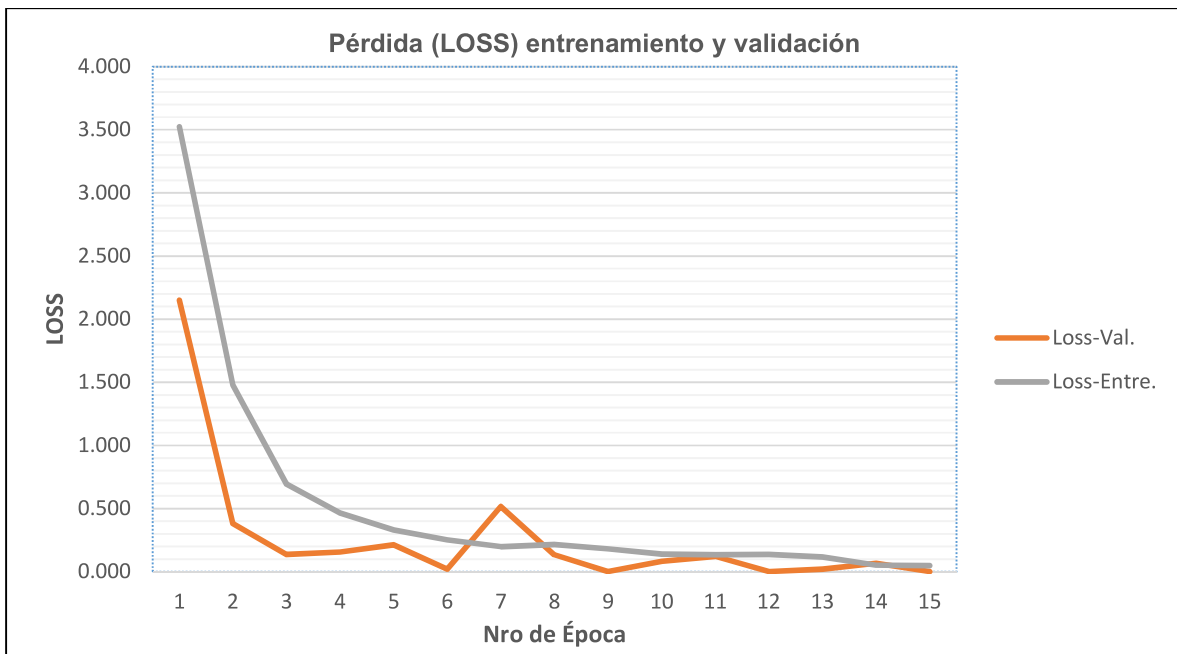


Figura 63. Loss de entrenamiento, tercer algoritmo, base CEDAR.

A continuación, se muestra la tabla de datos de la fase de pruebas del algoritmo red poco profunda para la base de datos CEDAR y después se presenta su matriz de confusión. Se puede observar que la exactitud de este modelo en la validación para la base CEDAR es de **96.36%** (tabla 21) y en la etapa de entrenamiento es de **98.36%** (tabla 20). Se puede observar una diferencia del 2%, esto indica que el algoritmo empezó una tendencia hacia un overfitting, sin embargo, como se explicó anteriormente se usó la técnica de EarlyStopping que permite detener el entrenamiento cuando el algoritmo ya no reduce su error (EMC) y empieza a tener un comportamiento hacia el overfitting.

MODELO POCOPROF, BASE CEDAR						
CLASES	PRES	SEN	ESP	TFP	F1	
1	1	0.8	1	0	0.8889	
2	1	1	1	0	1	
3	1	1	1	0	1	
4	1	1	1	0	1	
5	1	1	1	0	1	
6	1	1	1	0	1	
7	1	1	1	0	1	
8	1	1	1	0	1	
9	1	1	1	0	1	
10	1	1	1	0	1	
11	1	0.8	1	0	0.8889	
12	1	1	1	0	1	
13	1	0.8	1	0	0.8889	
14	1	1	1	0	1	
15	1	1	1	0	1	
16	1	1	1	0	1	
17	1	1	1	0	1	
18	1	1	1	0	1	
19	1	1	1	0	1	
20	1	0.8	1	0	0.8889	
21	1	1	1	0	1	
22	1	1	1	0	1	
23	1	0.8	1	0	0.8889	
24	1	1	1	0	1	
25	1	1	1	0	1	
26	1	1	1	0	1	
27	1	1	1	0	1	
28	1	1	1	0	1	
29	1	1	1	0	1	
30	1	1	1	0	1	
31	1	1	1	0	1	
32	1	1	1	0	1	
33	0.75	1	0.9954	0.0046	0.8571	
34	1	1	1	0	1	
35	1	1	1	0	1	
36	1	1	1	0	1	
37	0.25	1	0.9863	0.0137	0.4	
38	1	1	1	0	1	
39	1	1	1	0	1	
40	1	1	1	0	1	

41	1	1	1	0	1
42	1	1	1	0	1
43	1	1	1	0	1
44	0.75	1	0.9954	0.0046	0.8571
45	0.75	1	0.9954	0.0046	0.8571
46	1	0.5714	1	0	0.7272
47	1	1	1	0	1
48	1	1	1	0	1
49	1	1	1	0	1
50	1	1	1	0	1
51	1	1	1	0	1
52	0.75	1	0.9954	0.0046	0.8571
53	1	1	1	0	1
54	1	1	1	0	1
55	0.75	1	0.9954	0.0046	0.8571
Exact	0.9636	0	0	0	0
Prom	0.9636	0.9740	0.9993	0.0007	0.9610
DesvSt	0.1208	0.0794	0.0022	0.0022	0.0967

Tabla 21. Resultados fase de prueba, tercer algoritmo, base de datos CEDAR.

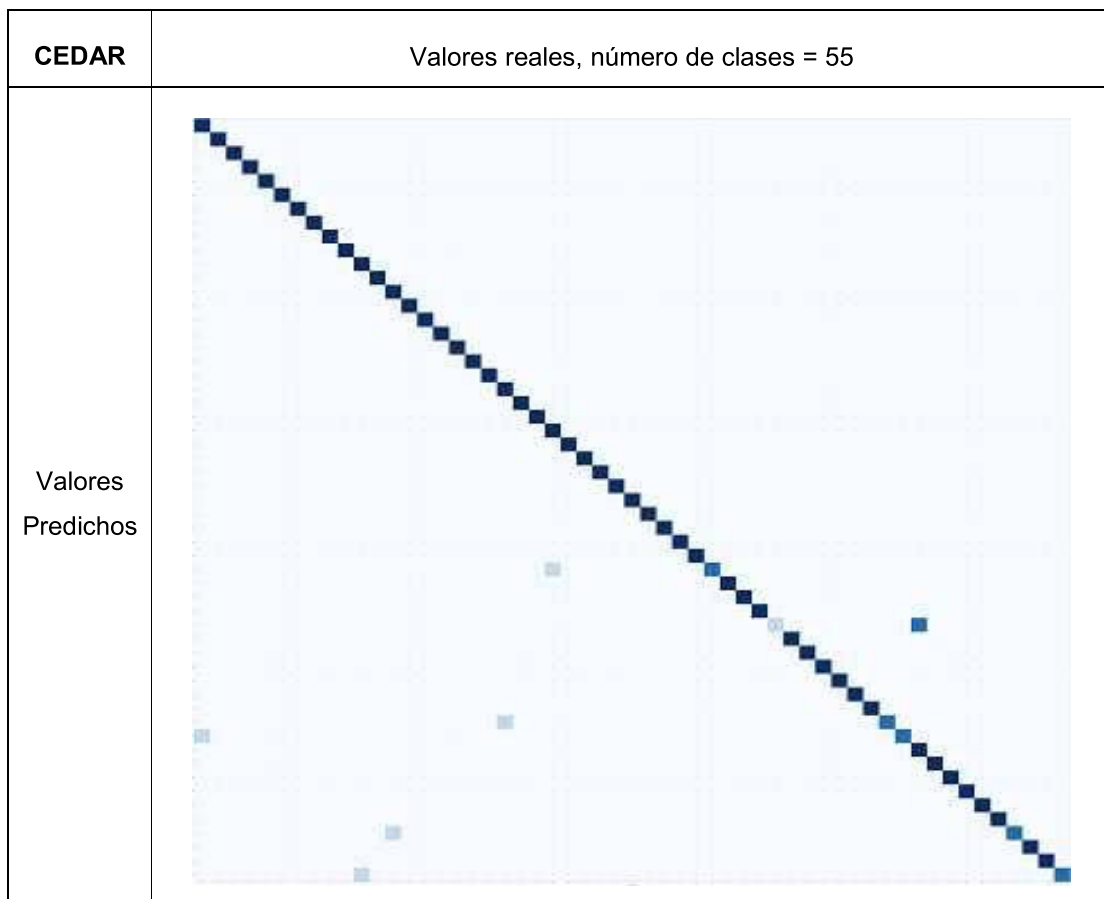


Figura 64. Matriz de confusión, tercer algoritmo, base CEDAR

En la figura 64, se muestra la matriz de confusión para el dataset CEDAR para el algoritmo reconocimiento de dígitos. Toda celda pintada de azul oscuro en la diagonal indica una validación al 100%, las celdas que están fuera de la diagonal, indican que en esa clase produjo una validación inferior al 99%.

4.1.6. Entrenamiento red convolucional poco profunda, base DB-GC.

Los datos de entrenamiento del tercer algoritmo muestran que en el entrenamiento de este algoritmo fue suficiente 11 épocas de entrenamiento para la base DB-GC. La exactitud que se alcanzó fue de **93.75%** y tomó un tiempo aproximado de dos horas y media completar el entrenamiento. Los datos de entrenamiento se muestran en la tabla 22 y en la figura 65, se muestra los datos de la exactitud de la fase de entrenamiento y en la figura 66, se muestra el error medio cuadrático (EMC).

Épocas	Exactitud		Error medio cuad(EMC)		Pérdida(loss)	
	Validac	Entrenam	Validación	Entrenam.	Loss-Val.	Loss-Entre.
1	0.26688	0.06960	0.00712	0.00803	2.86550	4.33793
2	0.58732	0.34297	0.00464	0.00653	1.88074	2.52361
3	0.70364	0.54092	0.00333	0.00491	1.06326	1.61375
4	0.80032	0.67860	0.00241	0.00362	1.03730	1.09216
5	0.83439	0.75321	0.00201	0.00289	0.20355	0.85629
6	0.84786	0.80341	0.00186	0.00234	0.86188	0.67677
7	0.88875	0.82624	0.00138	0.00207	0.89110	0.58129
8	0.88431	0.84157	0.00143	0.00187	0.71633	0.51757
9	0.88949	0.86744	0.00136	0.00159	0.57301	0.43262
10	0.93502	0.91097	0.00080	0.00110	0.36310	0.30383
11	0.93756	0.92997	0.00077	0.00087	0.85150	0.23421

Tabla 22. Datos de entrenamiento, tercer algoritmo, base DB-GC.

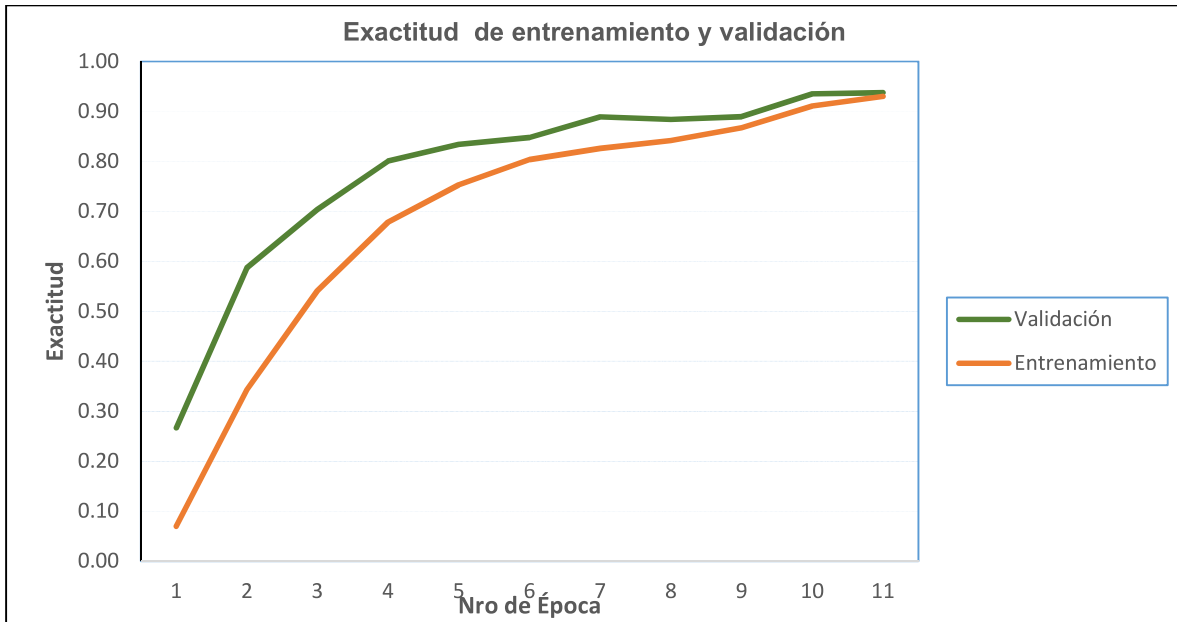


Figura 65. Exactitud, tercer algoritmo, base DB-GC.

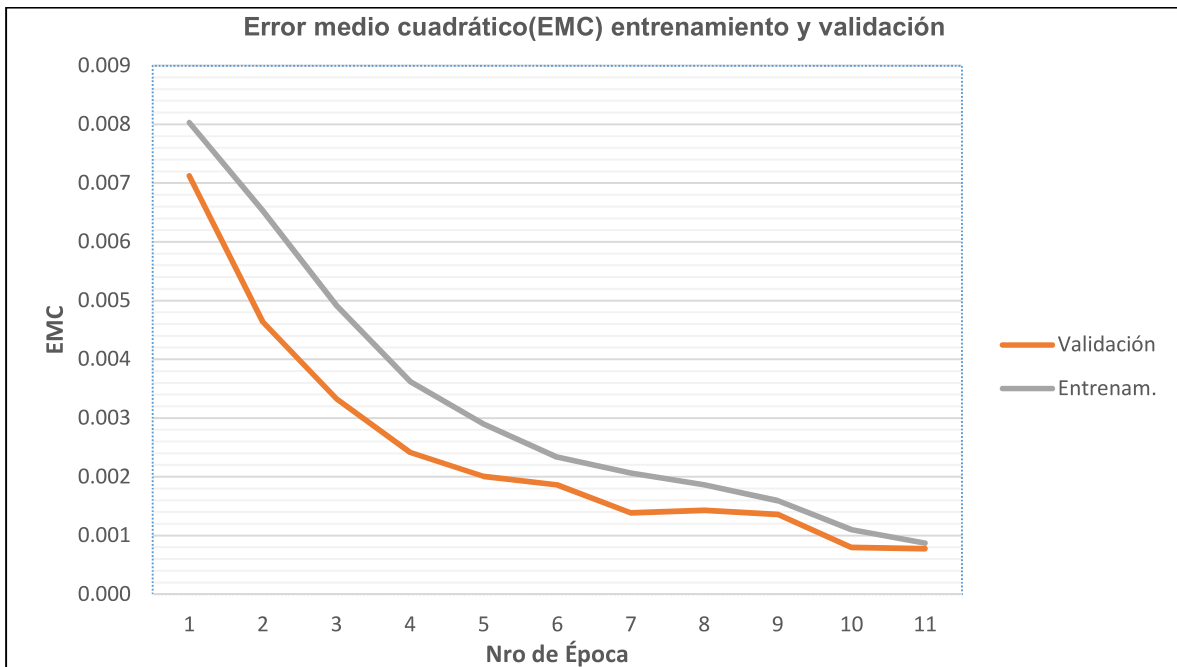


Figura 66. EMC, tercer algoritmo, base DB-GC.

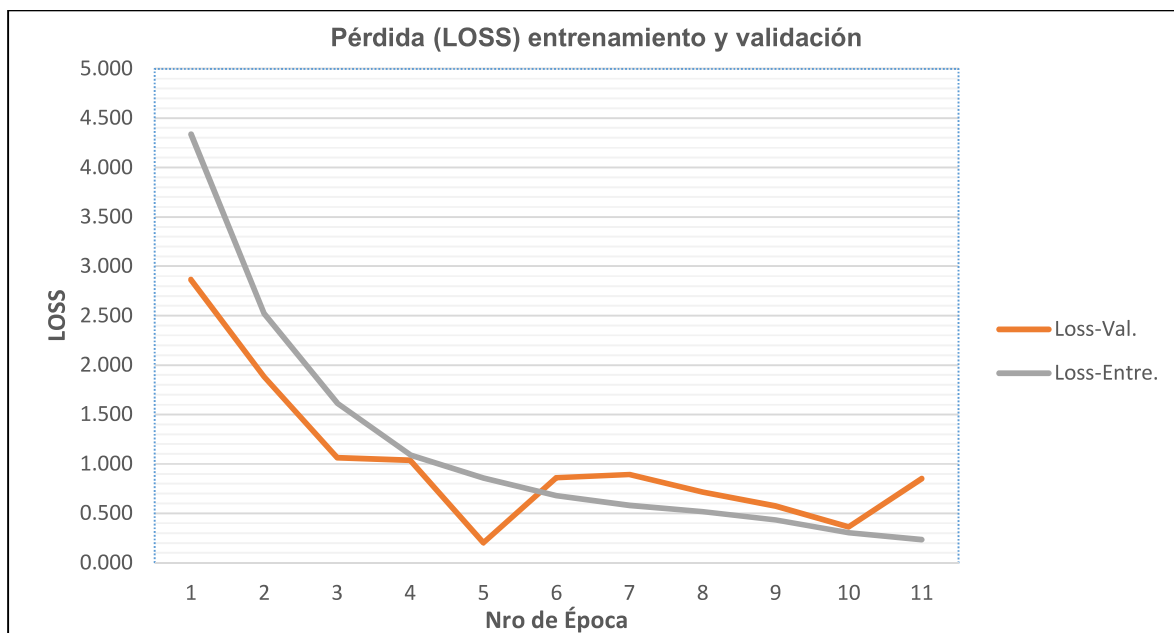


Figura 67. LOSS de entrenamiento, tercer algoritmo, base DB-GC.

En la figura 67 se muestra los valores de la función de pérdida para el algoritmo poco profundo en el dataset DB-GC.

Los datos de validación se muestran a continuación, la tabla 23 presenta los datos de validación, la exactitud de validación fue de **88.78%**.

Red-Poco Profunda DB-GC						F1-Score
CLASES	PRES	SEN	ESP	TFP		
1	1	0.7	1	0		0.8235
2	1	0.875	1	0		0.9333
3	1	1	1	0		1
4	1	0.7778	1	0		0.875
5	1	0.7778	1	0		0.875
6	0.8571	0.8571	0.9988	0.0012		0.8571
7	0.7143	0.8333	0.9976	0.0024		0.7692
8	0.7143	0.8333	0.9976	0.0024		0.7692
9	0.8571	0.75	0.9988	0.0012		0.8
10	0.8571	1	0.9988	0.0012		0.9231
11	1	0.7778	1	0		0.875
12	0.4286	0.6	0.9952	0.0048		0.5
13	0.8571	0.8571	0.9988	0.0012		0.8571
14	1	1	1	0		1
15	0.8571	1	0.9988	0.0012		0.9231
16	1	1	1	0		1
17	1	0.875	1	0		0.9333
18	0.8571	0.8571	0.9988	0.0012		0.8571
19	1	1	1	0		1
20	1	0.875	1	0		0.9333

21	1	0.875	1	0	0.9333
22	0.7143	1	0.9976	0.0024	0.8333
23	1	1	1	0	1
24	1	1	1	0	1
25	1	1	1	0	1
26	0.4286	0.75	0.9953	0.0047	0.5455
27	0.7143	0.625	0.9976	0.0024	0.6667
28	1	1	1	0	1
29	1	0.7	1	0	0.8235
30	1	1	1	0	1
31	0.8571	1	0.9988	0.0012	0.9231
32	0.8571	1	0.9988	0.0012	0.9231
33	1	0.5385	1	0	0.7
34	1	1	1	0	1
35	1	1	1	0	1
36	0.8571	1	0.9988	0.0012	0.9231
37	0	0	0.9917	0.0083	0
38	0.7143	1	0.9976	0.0024	0.8333
39	0.4286	1	0.9953	0.0047	0.6
40	0.8571	0.75	0.9988	0.0012	0.8
41	0.7143	1	0.9976	0.0024	0.8333
42	1	1	1	0	1
43	0.8571	1	0.9988	0.0012	0.9231
44	0.8571	1	0.9988	0.0012	0.9231
45	1	0.7778	1	0	0.875
46	1	0.875	1	0	0.9333
47	0.8571	1	0.9988	0.0012	0.9231
48	0.7143	1	0.9976	0.0024	0.8333
49	1	1	1	0	1
50	1	1	1	0	1
51	1	1	1	0	1
52	0.8571	1	0.9988	0.0012	0.9231
53	1	0.7	1	0	0.8235
54	1	1	1	0	1
55	1	1	1	0	1
56	0.8571	1	0.9988	0.0012	0.9231
57	0.8571	1	0.9988	0.0012	0.9231
58	0.8571	1	0.9988	0.0012	0.9231
59	0.7143	0.8333	0.9976	0.0024	0.7692
60	1	1	1	0	1
61	0.7143	0.7143	0.9976	0.0024	0.7143
62	0.7143	0.8333	0.9976	0.0024	0.7692
63	0.5714	1	0.9964	0.0036	0.7272
64	1	0.875	1	0	0.9333
65	0.7143	1	0.9976	0.0024	0.8333
66	0.7143	1	0.9976	0.0024	0.8333
67	1	1	1	0	1
68	0.8571	1	0.9988	0.0012	0.9231
69	0.5714	1	0.9964	0.0036	0.7272
70	1	1	1	0	1
71	0.8571	0.8571	0.9988	0.0012	0.8571
72	0.4286	1	0.9953	0.0047	0.6
73	0.7143	0.5	0.9976	0.0024	0.5882
74	1	0.875	1	0	0.9333
75	1	1	1	0	1
76	1	1	1	0	1

77	1	1	1	0	1
78	1	1	1	0	1
79	1	0.875	1	0	0.9333
80	1	0.875	1	0	0.9333
81	1	0.875	1	0	0.9333
82	1	1	1	0	1
83	1	0.6364	1	0	0.7778
84	0.5714	1	0.9964	0.0036	0.7272
85	1	0.7	1	0	0.8235
86	1	1	1	0	1
87	1	1	1	0	1
88	0.5714	1	0.9964	0.0036	0.7272
89	1	0.875	1	0	0.9333
90	1	1	1	0	1
91	1	1	1	0	1
92	1	1	1	0	1
93	0.5714	1	0.9964	0.0036	0.7272
94	1	0.7	1	0	0.8235
95	1	1	1	0	1
96	1	0.7	1	0	0.8235
97	1	0.6364	1	0	0.7778
98	1	0.5833	1	0	0.7368
99	0.5714	0.8	0.9964	0.0036	0.6666
100	1	0.5833	1	0	0.7368
101	1	1	1	0	1
102	1	1	1	0	1
103	1	1	1	0	1
104	1	0.875	1	0	0.9333
105	1	0.875	1	0	0.9333
106	0.8571	1	0.9988	0.0012	0.9231
107	0.5714	0.8	0.9964	0.0036	0.6666
108	1	0.875	1	0	0.9333
109	0.8571	1	0.9988	0.0012	0.9231
110	1	1	1	0	1
111	0.8571	0.75	0.9988	0.0012	0.8
112	0.7143	1	0.9976	0.0024	0.8333
113	1	0.875	1	0	0.9333
114	1	1	1	0	1
115	1	1	1	0	1
116	1	1	1	0	1
117	1	1	1	0	1
118	0.8571	0.8571	0.9988	0.0012	0.8571
119	1	1	1	0	1
120	1	1	1	0	1
121	1	1	1	0	1
Exact	0.8878	0	0	0	0
Prom	0.88783	0.90142	0.99906	0.00094	0.88261
DesvSt	0.17444	0.15202	0.00145	0.00145	0.14164

Tabla 23.Resultados fase de prueba, tercer algoritmo, base de datos DB-GC.

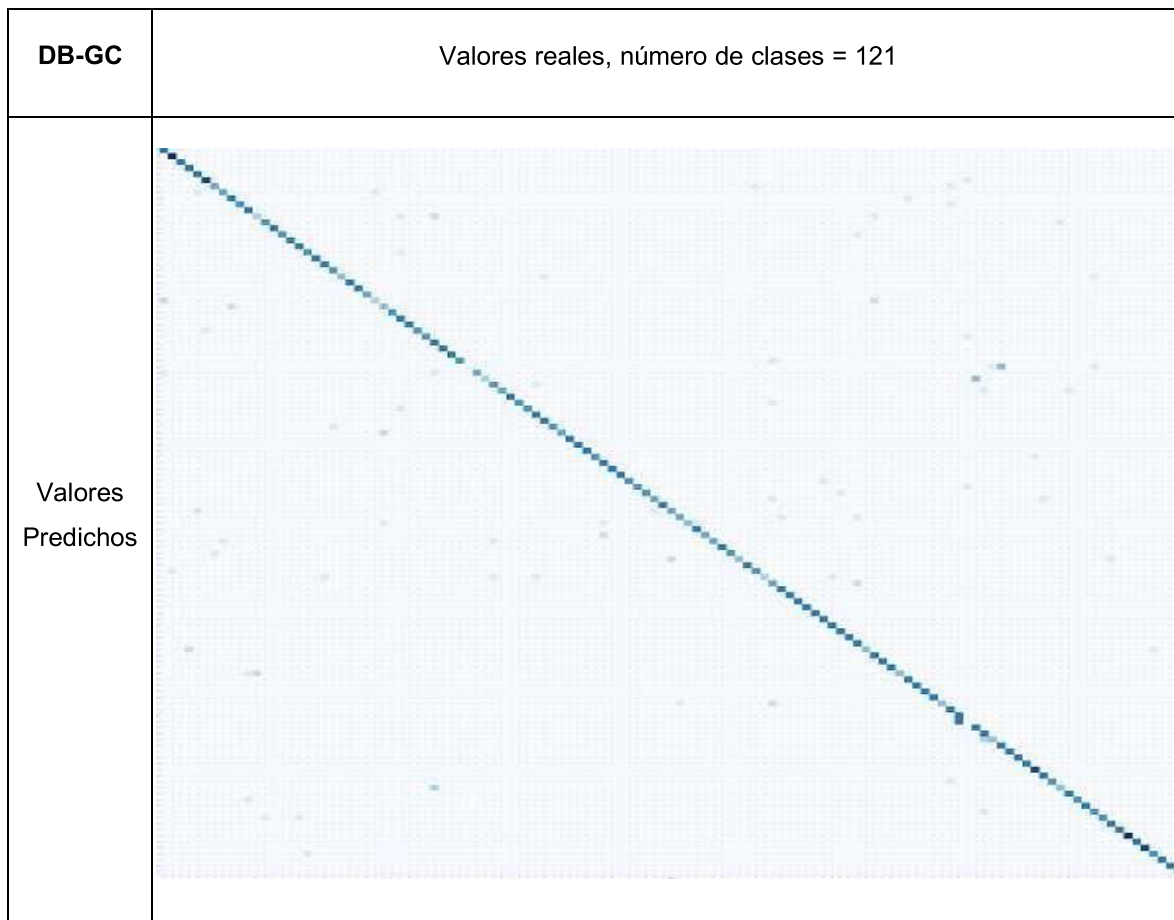


Figura 68. Matriz de confusión, tercer algoritmo, base DB-GC.

En la figura 68, se muestra la matriz de confusión para el dataset DB-GC para el algoritmo de red poco profunda, este dataset, tiene 121 clases. Cada celda pintada de azul oscuro en la diagonal indica una validación al 100%, las celdas que están fuera de la diagonal, indican que en esa clase produjo una validación inferior al 99%.

Para realizar la comparación de los tres algoritmos se procedió a evaluar la etapa de entrenamiento y la fase de pruebas, las métricas de la fase de validación son las más relevantes. Para la fase de entrenamiento se han tomado las métricas de exactitud, EMC, el tiempo necesario para completar el entrenamiento, el número de parámetros y el número de épocas, estos valores indicarán cual fue el modelo que se comportó mejor en el entrenamiento, estas medidas no son muy importantes, pero indican que tan eficaz es el algoritmo al ser implementado y entrenado. En la tabla 24, se muestran estos datos.

Entrenamiento		DB-GC	CEDAR
CN-GC	Exactitud(Acc)	97.46	99.73
	EMC	0.00030	0.00010
	Tiempo (h)	5.5	4
	Parámetros	507,205	490,213
	Epocs	18	15
	Firmas	4840	1320
Digitos Manuscritos	Exactitud(Acc)	82.26	95.19
	EMC	0.0020	0.0012
	Tiempo (h)	4	3
	Parámetros	229,699	224,089
	Epocs	27	26
	Firmas	4840	1320
CNN profunda Poco	Exactitud(Acc)	93.75	98.36
	EMC	0.00077	0.0005
	Tiempo (h)	2.5	1.5
	Parámetros	545,579	263,297
	Epocs	11	15
	Firmas	4840	1320

Tabla 24. Comparativo en la fase de entrenamiento.

La tabla 24, muestra que el primer algoritmo (CN-GC) es el de mejor exactitud, el algoritmo de red poco profunda tiene un mejor comportamiento en tres de las medidas escogidas (EMC, tiempo de entrenamiento y número de épocas), Se debe observar que para la base de datos CEDAR, la exactitud del algoritmo CN-GC, es ligeramente superior al algoritmo de red poco profunda, mientras que con la base DB-GC la exactitud del algoritmo CN-GC, es evidentemente superior a los otros.

4.2. RESULTADOS DE LA ETAPA DE PRUEBAS.

En la siguiente etapa de pruebas o validación, se usaron las siguientes medidas: la exactitud (accuracy) que debe ser complementada con la métrica de precisión, la sensibilidad (Recall), F1 score y finalmente la métrica de la tasa de falsos positivos, esta última medida debe ser lo más baja posible, o cero, puesto que es importante para este trabajo que, la firma de una persona sea clasificada correctamente y que esta decisión sea la más confiable posible.

Empezaremos con una comparación general en la base CEDAR, en donde se comparan las métricas exactitud, precisión, sensibilidad, especificidad, tasa de falsos positivos y F1. En la tabla 25, se presenta el promedio de las 55 clases, a excepción de la métrica exactitud (accuracy) porque este valor es único y general para cada modelo.

CN-GC	Accuracy		Precisión	Recall	Especif	TFP	F1
	0.9818	Promedio Desv St.	0.98181818 0.06492208	0.9854545 0.0519377	0.999665 0.001195	0.000335 0.001195	0.981527 0.045184
DÍGITOS	0.8273	Promedio Desv St.	0.82727273 0.24713231	0.8359982 0.2278538	0.996824 0.004528	0.003176 0.004528	0.811889 0.213247
POCO PROFUN	0.9636	Promedio Desv St.	0.9636364 0.1207757	0.9740255 0.0793829	0.9993327 0.0022115	0.0006673 0.0022115	0.9610400 0.0967428

Tabla 25. Comparativo general de los algoritmos, base de firmas CEDAR.

Los datos muestran que la mejor exactitud tiene el algoritmo CN-GC, siendo ligeramente superior comparado con el algoritmo de red poco profunda, el algoritmo de reconocimiento de dígitos manuscrito tiene la más baja exactitud.

Con relación a la base de datos DB-GC, la tabla 26 muestra que la mejor exactitud tiene el algoritmo CN-GC, siendo claramente superior comparado con el segundo y tercer algoritmo. En la figura 69, se aprecia visualmente lo expresado.

CN-GC	Accuracy		Precisión	Recall	Especif	TFP	F1
	0.9138	Promedio Desv.Std	0.913811 0.182258	0.934172 0.126872	0.999281 0.001515	0.000719 0.001515	0.907979 0.149151
DÍGITOS	0.7662	Promedio Desv.Std	0.766223 0.264244	0.774374 0.227373	0.998051 0.002192	0.001949 0.002192	0.751458 0.224261
POCO PROFUN	0.8878	Promedio Desv.Std	0.887832 0.174439	0.901422 0.152017	0.999061 0.001455	0.000939 0.001455	0.882607 0.141644

Tabla 26. Comparativo general de los algoritmos, base de firmas DB-GC.

La figura 69, muestra un comparativo de los tres algoritmos para los dos datasets usados (CEDAR y DB-GC) de la métrica de exactitud (accuracy).

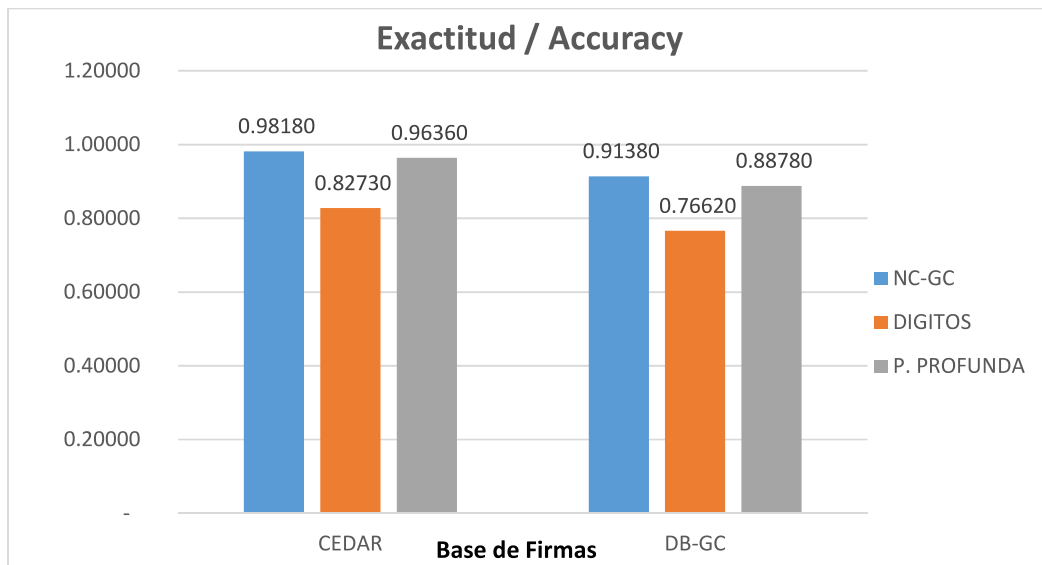


Figura 69. Comparativo métrica de exactitud/accuracy.

A continuación, en la figura 70, se muestra un gráfico comparativo de la precisión. Nótese que como estamos usando el promedio de la precisión, ésta presenta valores similares a la exactitud, un análisis más correcto de esta medida sería estudiar su comportamiento clase por clase.

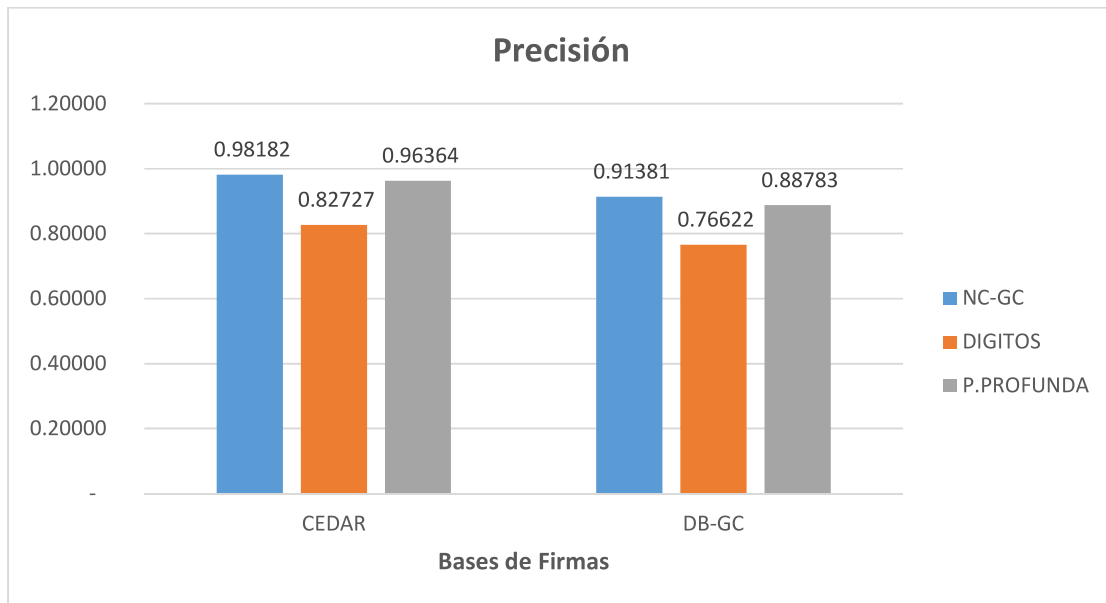


Figura 70. Comparativo de métrica de precisión.

Al comparar la precisión es importante tomar en consideración la tasa de falsos positivos (TFP) también conocida como FAR por sus siglas en inglés, esta medida debe ser lo más pequeña posible en este trabajo en particular. Interesa que la métrica TFP, que es aquella que indica que cuando un modelo predice que una firma pertenece a una persona y en realidad no es de esta persona sea muy baja o de ser posible cero, en este caso el primer algoritmo (CN-GC) tiene la más baja (0.000335) en la base de firmas CEDAR y 0.00072 en la base de firmas DB-GC, nuevamente se puede observar que el algoritmo CN-GC, tiene una mejor medida (tablas 25 y 26).

En la figura 71 se muestra el gráfico comparativo de TFP, en donde se puede observar que para las dos bases de firmas el primer algoritmo CN-GC, tiene la medida más baja, es decir que tiene un mayor grado de confianza, seguido del algoritmo (red neuronal poco profunda), el algoritmo de reconocimiento de dígitos manuscritos tiene la medida más alta es decir es el de más bajo rendimiento.

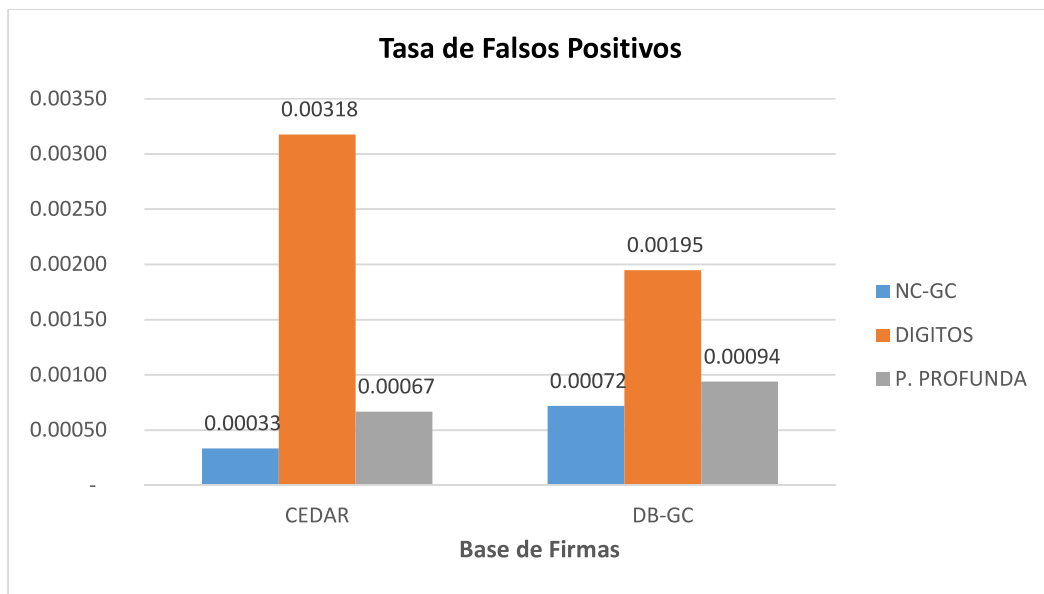


Figura 71. Comparativo de tasa de falsos positivos (TFP).

Con relación al indicador Recall, Se puede mencionar que este indica que tan sensible es el clasificador para encontrar instancias positivas, indica la probabilidad de clasificar correctamente la firma de una persona [26], interesa encontrar el algoritmo con mayor sensibilidad. El algoritmo CN-GC cumple con lo indicado, seguido muy de cerca del algoritmo la red neuronal poco profunda. En la figura 72 se muestra la métrica de sensibilidad.

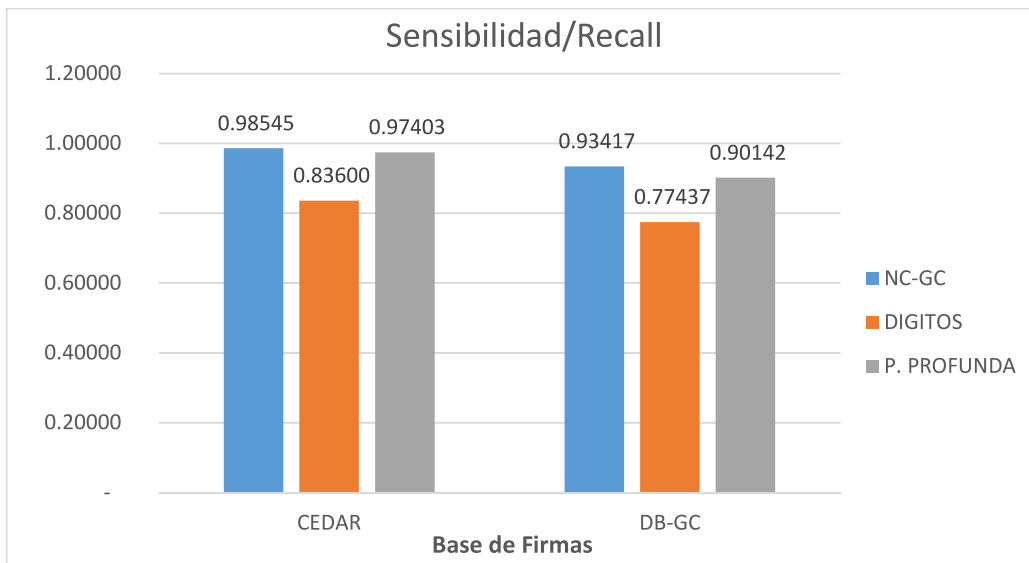


Figura 72. Comparativo métrica de sensibilidad.

En la figura 73 se presenta un comparativo de la métrica de especificidad para los tres algoritmos y los dos datasets. La especificidad es la probabilidad de que una firma que no pertenece a una persona, la predicción indique que no lo es. Esta medida, en nuestro estudio en particular no es muy relevante, normalmente es relevante cuando se analiza datos en el campo de la medicina, se la presenta como un dato informativo.

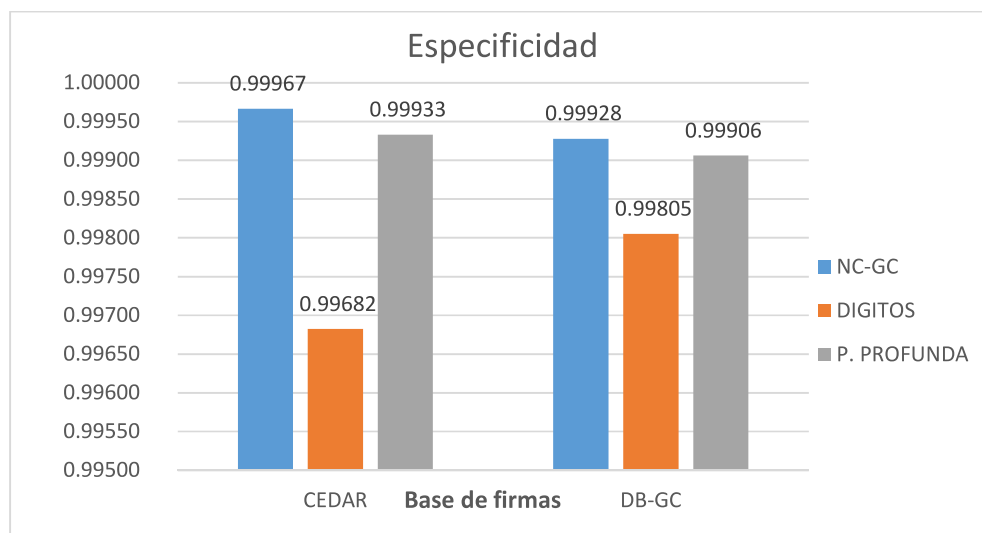


Figura 73. Comparativo métrica de especificidad.

F1-Score, es una medida que se usa cuando se necesita un balance entre precisión y sensibilidad. Se busca aquel modelo con un valor alto en esta métrica. Se puede observar que el algoritmo CN-GC tiene el valor más alto seguido muy de cerca del tercer algoritmo. En la figura 74, se muestra un comparativo de la métrica F1-Score.

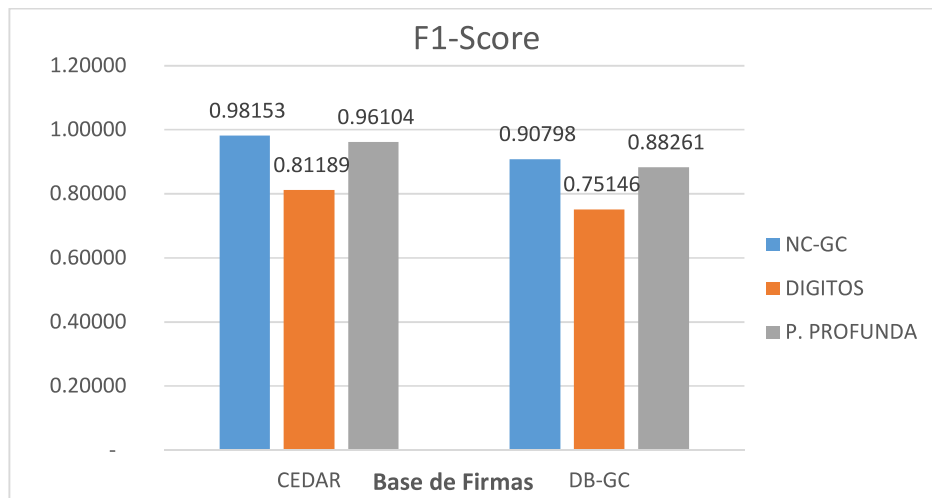


Figura 74. Comparativo F1-Score.

En conclusión el algoritmo CN-GC, en la fase de validación alcanza las mejores medidas, el tercer algoritmo está muy cerca mientras que el algoritmo de reconocimiento de dígitos es el de menor rendimiento esto se explica dado que este algoritmo no fue diseñado exclusivamente para firmas manuscritas, sin embargo se debe reconocer y dar mucho mérito este algoritmo (Dígitos manuscritos) tiene un buen rendimiento a pesar de no ser diseñado para este propósito, lo que sugiere que las redes neuronales convolucionales se adaptan muy bien en la clasificación de imágenes grises. No se puede comentar nada sobre imágenes en color ya que no era el alcance de este trabajo. En resumen, en la fase de entrenamiento el algoritmo de red convolucional poco profunda tiene las mejores medidas, mientras que en la validación el algoritmo CN-GC, tiene los mejores valores.

4.3. COMPARATIVO Y DISCUSIÓN DE RESULTADOS.

En la sección 4.2 se realizó un comparativo a nivel general de los tres algoritmos dando como resultado un mejor rendimiento del algoritmo CN-GC en la fase de validación, en base a los promedios de las métricas establecidas. En esta sección se hace un análisis del comportamiento de los algoritmos por cada clase o firmante. En la tabla 27, se presenta datos de la frecuencia y frecuencia acumulada de la métrica de precisión.

Clase	CN-GC		DIGITOS		POCO PROF	
	Frecuencia	% acum	Frecuencia	% acum	Frecuencia	% acum
100%	51	93%	33	60%	49	89%
75%	4	100%	15	87%	5	98%
50%	0	100%	6	98%	0	98%
25%	0	100%	1	100%	1	100%
0%	0	100%	0	100%	0	100%
T. CLASE	55		55		55	

Tabla 27. Datos de frecuencia y distribución acumulada de precisión base CEDAR.

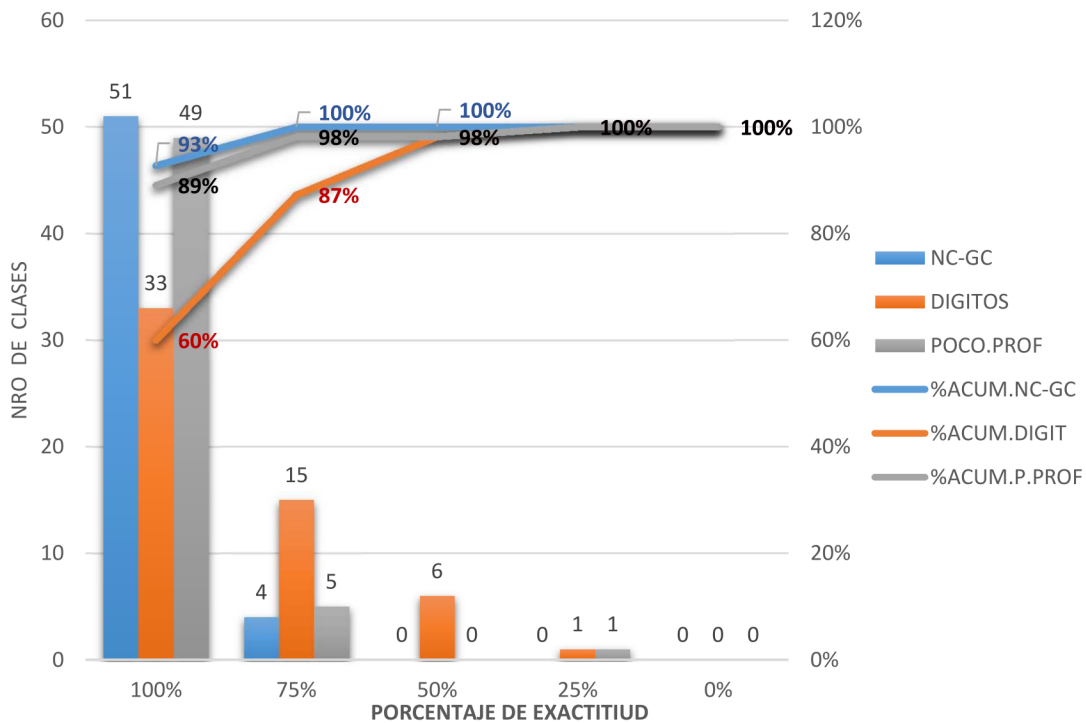


Figura 75. Histograma de frecuencias y distribución acumulada de precisión, CEDAR.

La figura 75, muestra un histograma frecuencia de la métrica de precisión en la base CEDAR, el gráfico de barras indica la cantidad de clases (firmantes) que fueron clasificados correctamente, con un 75%, 50%, 25% y 0% de precisión. El algoritmo CN-GC en 51 de 55 clases predijo correctamente todas las firmas de esa clase, Dígitos Manuscritos clasificó correctamente en 33 de 55 clases todas las firmas de estas clases y el algoritmo poco profundo clasificó correctamente en 49 de 55 clases. Los gráficos lineales representan la frecuencia acumulada, se puede observar que la frecuencia acumulada del algoritmo CN-GC está más arriba de los otros dos algoritmos lo que indica que en este algoritmo hay más clases clasificadas con una precisión del 100%, luego se puede observar el algoritmo de la red poco profunda y como último el algoritmo de reconocimiento de dígitos. En conclusión, la figura 75, dice que el algoritmo CN-GC tiene un mejor comportamiento de precisión clase por clase. También se puede observar en el gráfico que en el caso del algoritmo CN-GC solo existen clases con una precisión del 100% y del 75%, mientras que en los otros dos algoritmos existen clases con precisión del 50% y hasta del 25%, lo que muestra que el primer algoritmo tiene mayor precisión en la mayoría de clases. La curva de frecuencia acumulada del algoritmo de dígitos manuscritos es la que está debajo de las otras curvas, indicando que este algoritmo tiene la más baja precisión clase por clase.

En la tabla 28 se muestra el mismo análisis en la medida de precisión para la base de firmas DB-GC.

Clase	CN-GC		DÍGITOS		P.PROFUN	
	Frecuencia	% acum	Frecuencia	% acum	Frecuencia	% acum
100%	90	74.38%	46	38.02%	72	59.50%
86%	13	85.12%	27	60.33%	23	78.51%
71%	6	90.08%	12	70.25%	14	90.08%
57%	4	93.39%	17	84.30%	7	95.87%
43%	6	98.35%	8	90.91%	4	99.17%
29%	0	98.35%	6	95.87%	0	99.17%
14%	2	100.00%	1	96.69%	0	99.17%
0%	0	100.00%	4	100.00%	1	100.00%
TOTAL	121		121		121	

Tabla 28. Datos de frecuencia y distribución acumulada de precisión base DB-GC.

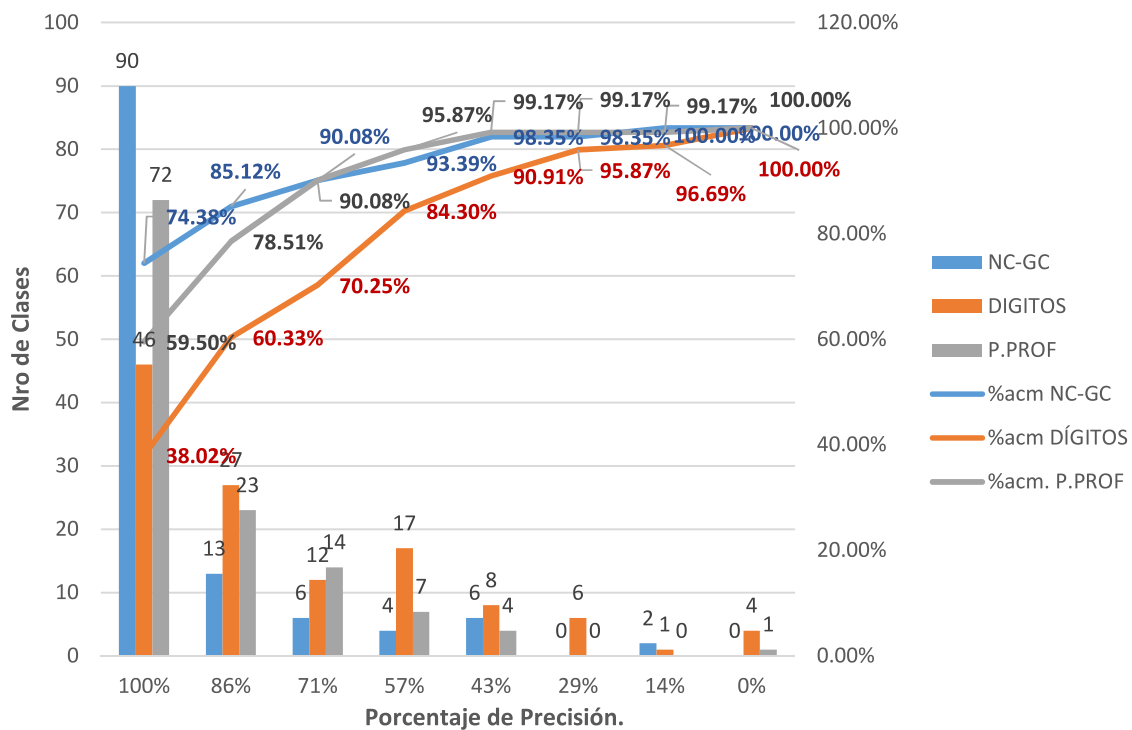


Figura 76. Histograma de frecuencias y distribución acumulada de Precisión. DB-GC.

La figura 76 muestra que el algoritmo CN-GC, tiene una precisión del 100% en una mayor cantidad de clases (90), seguido del algoritmo red poco profunda con una cantidad de 72 clases con precisión del 100%, en la curva de distribución acumulada tienden a comportarse de manera similar a partir del valor de precisión del 71%. El algoritmo de dígitos manuscritos está por debajo de las dos curvas indicando que tiene la menor precisión en todas las clases. Este gráfico permite concluir que el algoritmo CN-GC clasifica la mayor cantidad de clases con una precisión del 100% que los otros dos algoritmos.

La teoría de evaluación de modelos de redes neuronales dice que una alta precisión y una alta sensibilidad indican que tan bien un modelo maneja una clase, mientras más alto sea este valor, el algoritmo manejará de mejor manera esta clase, por lo que para verificar que tan correctamente se comportan los algoritmos en cada clase vamos a realizar un análisis de la métrica F1-Score, que es una métrica que combina las medidas de sensibilidad y precisión.

$$F1 = \frac{2 * Precision * Sensibilidad}{Precision + Sensibilidad} [26]$$

Los datos de la figura 77, son extraídos de la fase de validación de la base CEDAR, tablas 13, 17 y 21 correspondientes a los tres algoritmos.

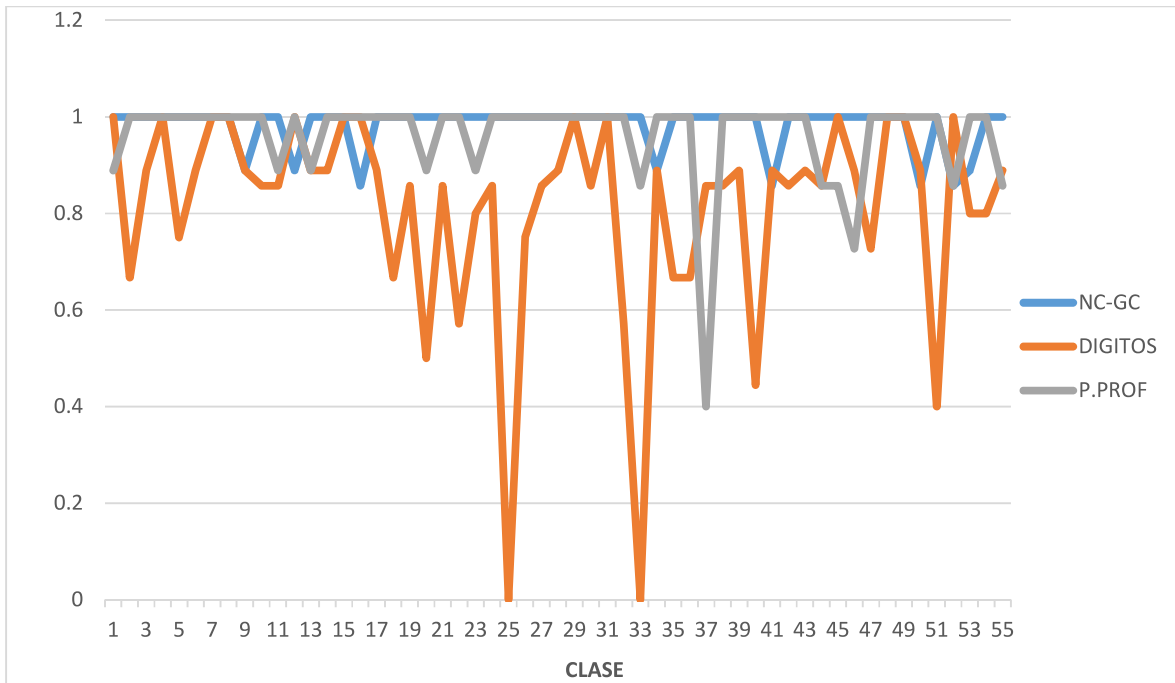


Figura 77. F1-Score, base de datos CEDAR

La figura 77, permite observar el comportamiento clase por clase en la base CEDAR de la métrica F1-Score que combina precisión y sensibilidad, mientras más alto sea este valor indica que el algoritmo maneja mejor esta clase. Se puede observar que el algoritmo CN-GC tiene muy pocos valores por debajo de 1 (clase 9, 13, 17, 33, 41 49), el algoritmo red neuronal poco profunda tiene más valores por debajo y el algoritmo de reconocimiento de dígitos es el de peor desempeño. Esto muestra que el algoritmo CN-GC tiene un mejor desempeño clase por clase que los demás algoritmos.

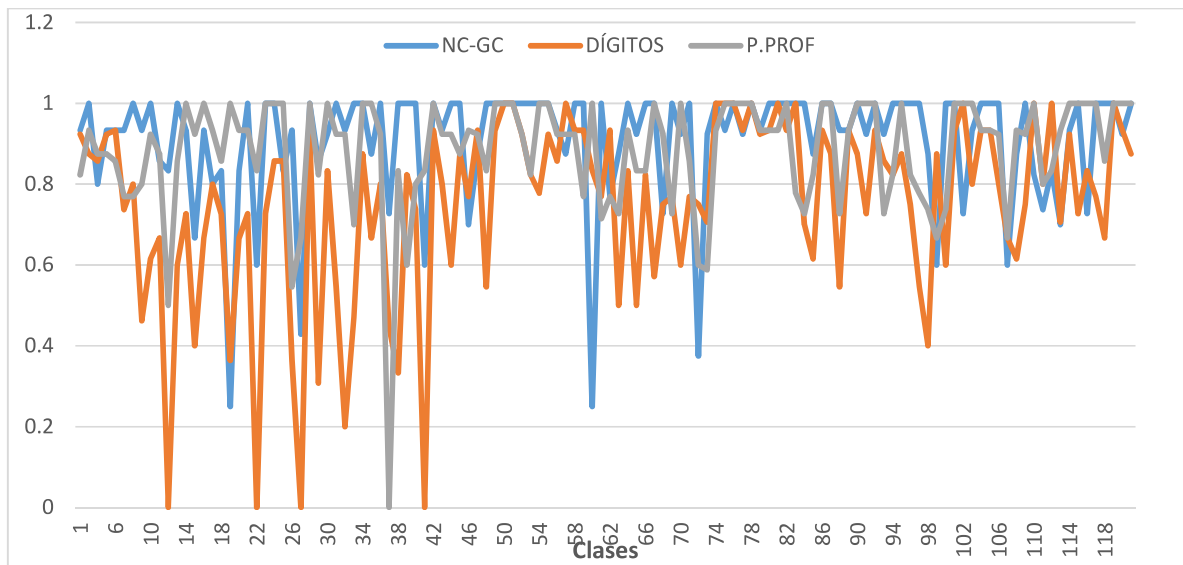


Figura 78. F1-Score, base de datos DB-GC.

En la figura 78, describe la métrica F1, de los tres algoritmos para la base DB-GC. Se puede observar que el primer algoritmo CN-GC tiene problemas al manejar 9 clases de las 121. También se puede observar que la red poco profunda tiene problemas en manejar más de 9 clases de las 121, el gráfico no permite apreciar con claridad por lo que se adjunta la siguiente tabla con datos extraídos de las tablas 15, 19, 23, tablas de la fase de validación para la clase DB-GC. Se realizará un análisis escogiendo al primer y tercer algoritmo, con valores de F1 inferiores a 0.7. No se ha tomado en cuenta el segundo algoritmo porque claramente se aprecia que es inferior a los otros dos.

CLASES	PRES	SEN	ESP	TFP	F1-Score
15	0.8571	0.5455	0.9988	0.0012	0.6667
19	0.1429	1	0.9929	0.0071	0.2501
22	0.4286	1	0.9953	0.0047	0.6
27	0.4286	0.4286	0.9952	0.0048	0.4286
41	0.4286	1	0.9953	0.0047	0.6
60	0.1429	1	0.9929	0.0071	0.2501
72	0.4286	0.3333	0.9952	0.0048	0.375
99	0.4286	1	0.9953	0.0047	0.6
107	0.4286	1	0.9953	0.0047	0.6

Tabla 29. Clases con F1-score menor a 0.7, algoritmo CN-GC, base DB-GC.

CLASES	PRES	SEN	ESP	TFP	F1-Score
12	0.4286	0.6	0.9952	0.0048	0.5
26	0.4286	0.75	0.9953	0.0047	0.5455
27	0.7143	0.625	0.9976	0.0024	0.6667
37	0	0	0.9917	0.0083	0
39	0.4286	1	0.9953	0.0047	0.6
72	0.4286	1	0.9953	0.0047	0.6
73	0.7143	0.5	0.9976	0.0024	0.5882
99	0.5714	0.8	0.9964	0.0036	0.6666
107	0.5714	0.8	0.9964	0.0036	0.6666

Tabla 30. Clases con F1-score menor a 0.7, algoritmo P. Prof, base DB-GC.

En las tablas 29 y 30, se aprecia los problemas en cada clase que tienen los algoritmos CN-GC y la red neuronal poco profunda respectivamente. Una alta precisión y una baja sensibilidad significan que un modelo no detecta muy bien una clase, pero que cuando lo hace es bastante confiable, en este caso estaría el algoritmo CN-GC en la clase 15 (tabla 29), mientras que la red poco profunda tiene problemas en la clase 27 y 73 (tabla 30). Un modelo con baja precisión y alta sensibilidad indica que el modelo detecta bien la clase, pero puede incluir firmas de otra clase, en este caso estarían las clases 19, 22, 41, 60, 99, 107 con el algoritmo CN-GC (tabla 29) y con la red poco profunda en las clases 39, 72, 99, 107 (tabla 30). Una baja precisión y una baja sensibilidad significan que el modelo no clasifica bien la clase y que esta predicción no es confiable, el algoritmo CN-GC tiene problemas en las clases 27, 72 (tabla 29) y la red poco profunda tiene problemas en las clases 12, 26, 37 (tabla 30). El algoritmo CN-GC y el tercer algoritmo se comportan de igual manera al analizar la métrica F1-Score.

En la tabla 31, se muestra los datos comparativos de la métrica de Tasa de Falsos positivos (TFP), esta medida debe tener un valor lo más bajo o de ser posible de ser cero en la mayoría de clases. La tabla 31 indica que el algoritmo CN-GC tiene problemas solo en 4 clases y que en 51 clases de la base CEDAR clasifico todas las firmas correctamente, el algoritmo de reconocimiento de dígitos manuscritos tiene problemas en 24 clases clasificando correctamente solo 30 clases y la red neuronal poco profunda tuvo problemas en 6 clases y en 49 clases todas las firmas fueron clasificadas correctamente. El grafico 79, presenta un comparativo de la tasa de falsos positivos mostrados en la tabla 31.

VALOR	CN-GC	CEDAR	
		DIGITOS	P.PROF
0.0046	4	13	5
0.0047	0	2	0
0.0092	0	5	0
0.0093	0	1	0
0.0137	0	1	1
0.0182	0	1	0
0.0183	0	1	0
TOTAL	4	24	6
%TFP	7.27%	43.64%	10.91%

Tabla 31. Tasa de falsos positivos base CEDAR.

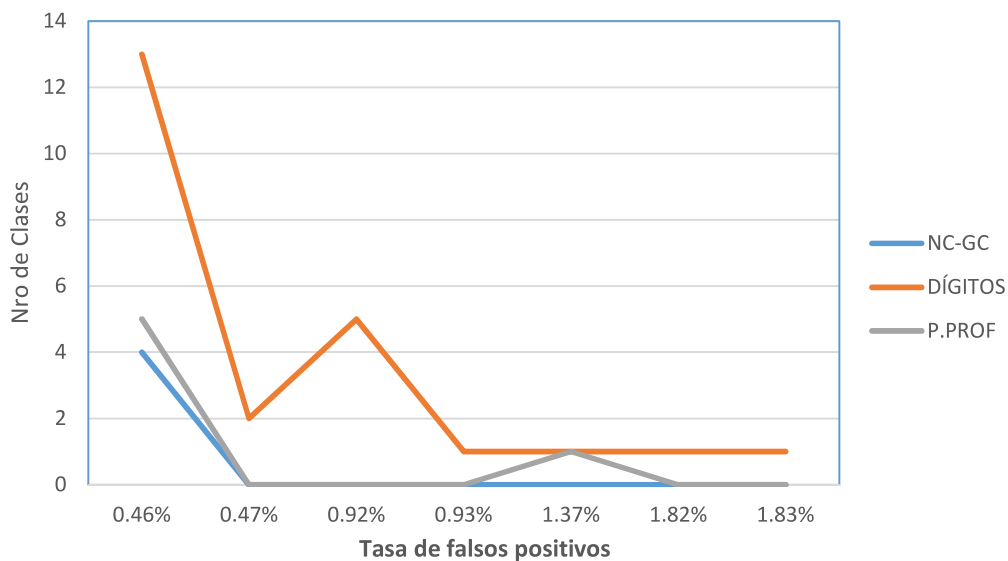


Figura 79. Comparativo tasa de falsos positivos, base CEDAR.

Al analizar los datos de la tabla 31, y de la figura 79, se puede apreciar que el algoritmo CN-GC y el algoritmo de red neuronal poco profunda se comportan de similar manera en la métrica de la tasa de falsos positivos con una ligera ventaja del algoritmo CN-GC en el punto 1.37. El algoritmo de reconocimiento de dígitos manuscritos está muy por encima eso significa que este algoritmo tiene una mayor probabilidad de provocar falsos positivos es decir de predecir que una firma pertenece a una persona pero en realidad no es de esta persona.

Seguidamente se presenta el mismo análisis de TFP de los algoritmos de clasificación en la base de firmas DB-GC.

VALOR	BASE DB-GC		
	CN-GC	DIGITOS	P.PROF
0.12%	13	27	22
0.24%	6	22	14
0.36%	4	17	6
0.47%	4	7	3
0.48%	2	1	1
0.59%	0	6	0
0.71%	2	1	0
0.83%	0	4	1
TOTAL	31	85	47
%TFP	25.62%	70.25%	38.84%

Tabla 32. Tasa de falsos positivos base DB-GC.

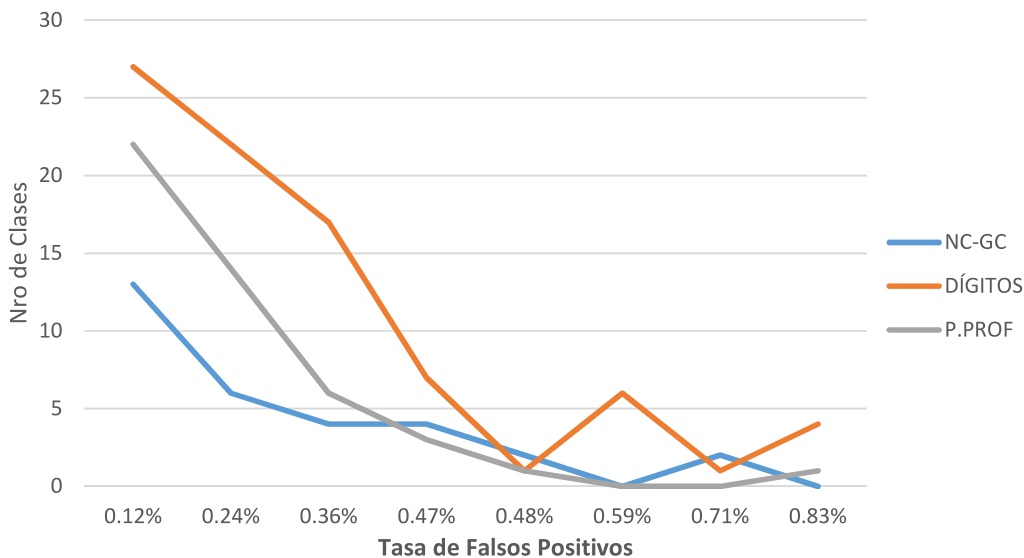


Figura 80. Comparativo de tasa de falsos positivos, base DB-GC.

En la tabla 32 y de la figura 80, se puede apreciar, el algoritmo CN-GC tiene la TFP más baja, con una ligera ventaja del algoritmo red neuronal poco profunda en los puntos 0.47, 0.48 y 0.71, sobre el algoritmo CN-GC. El algoritmo de reconocimiento de dígitos está muy por

encima eso significa que este algoritmo tiene una mayor probabilidad de provocar falsos positivos.

Como conclusión se puede afirmar, que el algoritmo CN-GC y el algoritmo red neuronal poco profunda tiene un comportamiento similar, con ligeras ventajas del algoritmo CN-GC sobre el algoritmo de la red neuronal poco profunda con la base de firmas CEDAR, en la base de firmas DB-GC el algoritmo CN-GC tiene un comportamiento más significativo que el algoritmo de la red neuronal poco profunda. El algoritmo de reconocimiento de dígitos está por debajo de los otros dos algoritmos en las dos bases de firmas, sin embargo, es meritorio que este algoritmo se comporte bastante bien clasificando firmas manuscritas, a pesar de que su diseño fue para clasificar dígitos manuscritos. En la fase de entrenamiento evidentemente el tercer algoritmo es más liviano que los otros dos, su entrenamiento tiene una duración menor que el primer y segundo algoritmo.

5. CONCLUSIONES Y RECOMENDACIONES.

De este trabajo de investigación se pueden sacar conclusiones respecto a las bases de datos utilizadas y a un análisis comparativo de los algoritmos implementados.

5.1. CONCLUSIONES.

Para este trabajo se usó dos bases de datos CEDAR y DB-GC, la segunda base de datos fue levantada de manera manual, se recolectaron firmas de 121 personas, con 45 firmas por cada firmante. La emergencia sanitaria relacionada con el COVID complicó la recolección personal, por lo que se decidió por una recolección vía internet para ciertos firmantes, en donde era el usuario quién digitalizaba la imagen y luego enviaba por correo, esto hizo que las firmas no fueran claras, que no estén capturadas con la misma resolución, entre otras, provocando que la base DB-GC no sea procesada como la base de datos CEDAR. El uso de diferentes tipos de papel hizo que las firmas de la base DB-GC no tengan el mismo pre-proceso de la imagen, adicionalmente el cambio de formato en el documento de recolección también provocó problemas en la fase del pre-proceso. A pesar de todos los inconvenientes presentados se consiguió en el algoritmo CN-GC y el algoritmo red neuronal poco profunda una exactitud superior al 91%, y superior al 88% respectivamente, logrando probar que si es posible clasificar firmas con un error inferior al 12%. Con relación al algoritmo de reconocimiento de dígitos cuyo objetivo no fue reconocer firmas manuscritas, se puede mencionar que tuvo un excelente desempeño superando el 75% de exactitud con la base DB-GC y el 85% con la base CEDAR, todo esto permite concluir que las redes neuronales convolucionales se pueden adaptar eficazmente en la clasificación para las imágenes en tonos de gris.

Existe el paradigma para las redes neuronales, de la necesidad de un gran volumen de datos para realizar un buen entrenamiento en una red neuronal, la base de datos CEDAR tiene 24 firmas por cada firmante y la base de datos DB-GC tiene 45 firmantes en la mayoría de casos y un mínimo de 30 firmas en pocos firmantes, con lo que se logró procesar 166 firmantes, un número relativamente pequeño, sin embargo este número de datos combinado con la técnica de aumento de datos que ofrece la librería Keras, hizo que los modelos se entrenaran con una exactitud aceptable con el primer y tercer algoritmo, para las dos bases de firmas. Esto permite concluir que no necesariamente se necesita un enorme volumen de datos para clasificar firmas manuscritas usando redes neuronales convolucionales y que esto depende del tipo de clasificación que se desee realizar.

Otro paradigma en las redes neuronales convolucionales es que un modelo debe ser complejo en su capa de extracción de características y de clasificación. El algoritmo red neuronal poco profunda es una muestra de que la red neuronal convolucional no necesariamente debe ser compleja, llena de capas convolucionales, lleno de un gran número de filtros, para lograr un buen modelo de clasificación de imágenes. Adicionalmente la parte clasificadora de este modelo solo tiene dos capas, que son suficientes para una buena clasificación. Esto muestra que la red convolucional puede ser pequeña, con un pequeño número de capas convolucionales y de clasificación para lograr buenos modelos de clasificación para imágenes en todos de gris.

Para lograr una buena clasificación en imágenes grises es necesario que la imagen contenga una cantidad de información aceptable, el hecho de reducir la imagen como fue el caso del uso de la técnica de esqueletización provoca que la imagen pierda características y la clasificación no sea efectiva. El uso de la técnica de esqueletización de una firma no fue de ayuda, con los modelos usados no se logró superar el 3% de exactitud, un rendimiento en los tres modelos muy pobre, por lo que se concluye que el uso de esta técnica no ayudó en nada a la clasificación de imágenes con estos tres modelos usados.

Como conclusiones finales se puede mencionar que:

Si es posible usar un modelo de red neuronal convolucional, para identificar firmas manuscritas con una exactitud superior al 88%, o lo que es lo mismo un error inferior al 12%, que el éxito de un modelo de clasificación de firmas manuscritas, depende mucho de la calidad de los datos tanto en la recolección y en la etapa de pre-proceso.

Los modelos comparados son algoritmos no muy complejos ni pesados, en especial el modelo de red neuronal poco profunda, estos podrían ser implementados en computadores de uso común, inclusive en dispositivos de mano tales como una tableta personal, un teléfono celular inteligente, lo que contribuye a la creación de herramientas de bajo costo, las mismas que pueden ser usadas por instituciones pequeñas, que no tengan mucha capacidad económica para la inversión en tecnología.

5.2. RECOMENDACIONES.

Con relación a la recolección de información, se debe estandarizar el tipo de papel en donde se registran las firmas, se sugiere que sea un papel de gramaje grueso, se pudo observar que

el papel delgado ocasiona problemas al momento de digitalizar la firma con un scanner, debido a que se genera ruido excesivo en la imagen digitalizada. Adicionalmente se debe mejorar el formato de recolección de firmas para no provocar el registro de la firma en áreas de la hoja que no son deseados, de esta manera se puede obtener imágenes de firmas más uniformes y válidas.

El alcance del este proyecto implica el uso de algoritmos de clasificación supervisada, de todos los existentes, se ha centrado en el estudio de las redes neuronales convolucionales. Debido a que es clasificación supervisada y debido al deseo del autor en profundizar en el conocimiento de este tipo de red neuronal, esto implica que se debe tener un número de clases (firmantes) establecidas, definidas, fijas para su entrenamiento y posterior uso, pero en la práctica en una institución las clases (firmantes) no permanecen fijas, éstas van creciendo cada día puesto que los clientes o socios de la institución se registran diariamente, no sería eficiente entrenar el algoritmo a usar cada día con las clases (firmantes) existentes más las nuevas registradas. Se propone un futuro proyecto de investigación, realizar la misma clasificación, pero de manera no supervisada o usar un método de transferencia de aprendizaje, que se concentre en el entrenamiento de la capa clasificadora, para lograr disminuir drásticamente el tiempo de entrenamiento.

El uso de estos algoritmos puede ser muy útil en la verificación de firmas manuscritas en cheques, actualmente en la mayoría de instituciones bancarias la firma del cheque es verificada de manera visual con la base de datos de firmas por parte del encargado de validar el documento, como se explicó en la introducción de este trabajo, cada banco debe procesar una gran cantidad de cheques diariamente, para proceder a la aceptación o rechazo del mismo, por lo que es crítico la rapidez y confiabilidad de este proceso. Se puede extender un futuro trabajo a la implementación de estos algoritmos en el área de la cámara saliente de compensación de una institución bancaria.

REFERENCIAS

- [1] M. Miah, M, Yousuf, S. Mia., «Handwritten Courtesy Amount and Signature Recognition on Bank Cheque using Neural Network,» *International Journal of Computer Applications*, vol. 118, nº 5, pp. 20-26, May 2015.
- [2] R. Palacios, A. Gupta., «A system for processing handwritten bank checks automatically,» *Image and Vision Computing*, vol. 26, nº 10, pp. 1297-1313, 2008.
- [3] P.Agrawal, D. Chaudhary, V. Madaan, A. Zabrovskiy, R. Prodan, D. Kimovski, C.Timmerer, «Automated bank cheque verification using image processing and deep learning methods,» *Multimedia Tools and Applications*, vol. 1, nº 1, pp. 1-32, October 2020.
- [4] R. Acosta, *Verificación de firmas*, Bogotá: Universidad Nacional de Colombia, 2013.
- [5] R. Jayadevan, S. R. Kolhe, P. M. Patil, U. Pal, «Automatic processing of handwritten bank cheque images a survey,» *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 15, nº 4, pp. 267-296, June 2011.
- [6] V. Malekian, A. Aghaei, M. Rezaeian, M. Alian, «Rapid off-Line Signature Verification Based on Signature Envelope and Adaptive Density Partitioning.,» *First Iranian Conference on Pattern Recognition and Image Analysis (PRIA)*, 2013.
- [7] S. Chandra, S. Maheskar, «Offline signature verification based on geometric feature extraction using artificial neural network,» *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pp. 410-414, 2016.
- [8] L. Hafemann, R. Sabourin, L.Oliveira, «Offline Handwritten Signature Verification,» *Seventh International Conference on Image Processing Theory - Literature Review, Tools and Applications (IPTA)*, pp. 1-8,, 2017.

- [9] O. Rosso, R. Ospina, A. Frery, «Classification and Verification of Handwritten Signatures with Time Causal Information Theory Quantifiers.,» *Plos One*, vol. Vol. 11., nº 12., 2016.
- [10] D. Ahmedt, E. Delgado, J. Vargas, «Generación de espacios de representación de firmas dinámicas : Una revisión enfocada al análisis de complejidad,» *Scientia Et Technica.*, vol. XVI, nº 48, pp. 244-252, 2011.
- [11] S.N. Gunjal, B.J. Dange, A.V. Brahmane, «Offline Signature Verification using Feature Point Extraction,» *International Journal of Computer Applications*, vol. 141, nº 14, pp. 6-12, 2016.
- [12] E. Roe, C. Mello, «Binarization of color historical document images using local image equalization and XDoG,» de *International Conference on Document Analysis and Recognition*, 2013.
- [13] P. Kandhway, A. Bhandar, A. Singh, «A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization,» *Biomedical Signal Processing and Control*, vol. 56, nº 1, Septiembre 2019.
- [14] M. Villa, S. Yáñez, R. Enrique, «Fundamentos de la reducción de ruido en imágenes,» *Jovenes en la Ciencia*, vol. 3, nº 2, pp. 2531-236, 2017.
- [15] K. Sánchez, F. Garcia, *Procesamiento Digital de Señales*, Ciudad de México: Universidad nacional Autonoma de México, 2011.
- [16] L. Boudaoud, B. Solaiman, A. Tari, «A modified ZS thinning algorithm by a hybrid approach,» *The Visual Computer*, vol. 34, nº 5, pp. 689-706, 2017.
- [17] G. Narkhede, D. Patil, «Review of Feature Extraction Techniques for Offline Signature Verification,» *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, nº Issue 2, pp. 1569-1571, Febrero 2014.
- [18] J. Salas, D. Moriñigo, F. Espí, «Desarrollo con matlab de una Red Neuronal para estimar la demanda de energía eléctrica,» Valladolid, Universidad de Valladolid, 2016, pp. 1, 100.

- [19] P. Ponce, *Inteligencia Artificial con Aplicaciones para la Ingeniería*, 1era ed., Ciudad de México: Grupo AlfaOmega, 2010.
- [20] E. Serna, «Desarrollo e innovación en ingeniería.,» *Serie Ingeniería y Ciencia*, vol. 1, pp. 173-183, 2017.
- [21] D. Cireşan. U. Meier, J. Masci, L. Gambardella, J. Schmidhuber, «Flexible, High Performance Convolutional Neural Networks for Image Classification,» *Twenty-second international joint conference on artificial intelligence.*, vol. 1, nº 1, pp. 1236-1242, June 2011.
- [22] F. Giménez-Palomares, J. A. Monsoriu, E. Alemany-Martínez, «Aplicación de la convolución de matrices al filtrado de imágenes.,» *Modelling in Science Education and Learning*, vol. 9, nº 2 , pp. 97-107, Enero 2016.
- [23] E. Benavidez, W. Fuertes, S. Sánchez-Gordón, L. Tello., «Deep Learning Techniques in Early Detection of Phishing Attacks: A Survey,» *En espera de ser publicado*, vol. 1, nº 1, pp. 1-45, Agosto 2020.
- [24] T. Fawcett, «An introduction to ROC analysis,» *Pattern recognition letters*, vol. 27, nº 1, pp. 861-874, 19 December 2005.
- [25] M. Hanmandlu, A. Sronothara, S. Vasikarla., «Deep Learning based Offline Signature Verification,» *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018*, vol. 1, nº 1, pp. 732-737, 2018.
- [26] A. Ceballo, E. Borland, S. Steria, J. Solé, «Usando técnicas de visión por computador para la validación de firmas manuscritas,» *Universidad politécnica de Catalunya*, vol. 1, nº 1, pp. 1-52, 4 Julio 2019.
- [27] M. Okawa, «Synergy of foreground-background images for feature extraction: Offline signature verification using Fisher vector with fused KAZE features,» *Pattern Recognition* , vol. 79, nº Tokyo, pp. 480-487, 2018.

- [28] A. Beresneva, A. Epishkina, D. Shingalova, «Handwritten signature attributes for its verification.,» *National Research Nuclear University MEPhI*, vol. 1, n° 1, pp. 1477-1481, 2018.
- [29] O. Mohamed, I. El-Feghi, «Angle Features Extration of Handwritten Hignatures,» *Conference: 2015 International Conference on Computer Vision and Image Analysis Applications*, vol. 1, n° 1, Enero 2015.
- [30] R. Zouari, R. Mokni, M. Kherallah, «Identification and verification system of offline handwritten signature using fractal aproach,» *Internacional image processing applications and system conference 2014*, vol. 1, n° 1, pp. 1-4, 2014.
- [31] S.Ooia, A.Teohb, Y. Pang, B. Hiew, «Image-based handwritten signature verification using hybrid methods of discrete Radon transform, principal component analysis and probabilistic neural network,» *Applied Soft Computing Journal*, vol. 40, n° 1, pp. 274-282, 2016.
- [32] M.S. Shirdhonkar, M.B. Kokare, «Off-Line Handwritten Signature Retrieval,» *ACM*, vol. 1, n° 1, pp. 290-294, 2012.
- [33] B. Zhang, «Offline signature verification and identification by hibrid feactures and Suport Vector Machine,» *Artificial Inteligence and Soft Computing*, vol. 2, n° 4, pp. 302-319, 2011.
- [34] S. Wang, S Jia, «Signature handwriting identification based on generative adversarial networks,» *Journal of Physics*, vol. 1, n° 1, pp. 1-5, 2019.
- [35] L. Hafemann, R.Sabourin, L. Oliveira, «Writer-independent Feature Learning for Offline Signature Verification using Deep Convolutional Neural Networks,» de *IJCNN 2016*, Quebec, 2016.
- [36] S. Dey, A. Dutta, J. Toledo, K. Ghosh, J. Lladós, U.Pal, «Signet: Convolutional Siamese Network for Written independent Offline Signature Verification,» *Pattern Recognition Letters (preprint)*, vol. 1, n° 1, pp. 1-7, 2017.

- [37] J. Anamika, S. Satish Kumar, S. Krishna Pratap, «Handwritten signature verification using shallow convolutional neural network.,» *Multimedia Tools and Applications*, vol. 79, nº 1, pp. 1-23, 7 April 2020.
- [38] C. Bernal, D. Salvarrieta, T. Sánchez, R. Salazar, Metodología de la Investigación, Segunda Edición en Español ed., vol. 1, F. H. Carrasco, Ed., México, México: Pearson Educación, 2006, pp. 1-304.
- [39] S. Pratt, A. Ochoa, M. Yadaw, A. Sheta, M. Eldefravy, «Handwritten Digits Recognition Using Convolutional Neural Networks,» *The Journal of Computing Sciences in Colleges*, vol. 34, nº 5, pp. 40-46, 5 April 2019.
- [40] T. Rabbi, S. Rahman, P. Biswash, J.Kim, A.Sheikh, A.Saha, M.Uddin, «Handwritten Signature Verification Using CNN with Data Augmentation.,» *Journal of Contents Computing*, vol. 1, nº 1, pp. 25-37, December 2019.
- [41] M.Yapıcı, A.Tekerek, N. Topaloğlu., «Deep learning-based data augmentation method and signature verification system for offline handwritten signature,» *Pattern Analysis and Applications (2020)*, vol. 16, nº 4, pp. 1-15, September 2020.
- [42] L. E. Vílchez-Fernández, «Revisión de algoritmos de verificación automática de firmas off-line.,» *Interfases*, vol. 0, nº 10, pp. 149 -163, 2017.